

Die Berechnung einer Epidemie als *computational document*

Jens Wendelstorf

PD Dr. rer.nat.habil. J. Wendelstorf,
Technische Universität Clausthal, D-38678 Clausthal-Zellerfeld,
E-Mail: jens.wendelstorf@tu-clausthal.de
Publiziert als Hochschulschrift unter DOI:10.21268/20200921-5
(c) 2020 jens@wendelstorf.de

Vorbemerkung:

Dieses Dokument wurde für den Einsatz in der Lehre verfasst. Die enthaltenen Programme erfordern Kenntnisse der Wolfram Language, können jedoch ohne Vorkenntnisse benutzt werden um selbst mit den interaktiven Modellen zu experimentieren. Ziel ist eine Heranführung an die wissenschaftliche Denkweise und das computational thinking. Dazu ist genügend Aufmerksamkeit und eine gewisse Komplexitätstoleranz aufzubringen. Um das Verhalten von komplexen Systemen zu verstehen und Entscheidungen auf der Basis von wissenschaftlichen Modellen treffen zu können ist eine gewisse eigene Erfahrung auf diesem Gebiet erforderlich!

Sie lesen gerade die Druckversion:

Arbeiten Sie besser mit dem computational document, indem Sie die an die PDF Datei angehängte Datei `Epidemie.nb` mit dem Wolfram Player (<https://www.wolfram.com/player/>) öffnen. Um die Programme auch selbst ändern zu können benötigen Sie die Software Mathematica, wie sie zum Beispiel beim Raspberry Pi Computer mitgeliefert wird.

Kurzfassung (Abstract):

*Die SARS-CoV-2 Pandemie hat die Epidemiologie in den Fokus der öffentlichen Aufmerksamkeit gerückt. Dazu ist ein grundlegendes epidemiologisches Verständnis erforderlich. Diesem Verständnis der grundlegenden Zusammenhänge dient dieses **Paper**.*

*Dazu wird **das einfachste epidemiologische Modell** herangezogen, welches die wichtigsten Phänomene beinhaltet. Mit diesem **computational document** wird dem Leser die Möglichkeit gegeben, alle Berechnungen selbst nachzuvollziehen und ein grundlegendes Verständnis für den Verlauf und die Beeinflussung einer Epidemie zu erwerben. Zudem wird ein Einblick in die Denk- und Arbeitsweise der wissenschaftlichen Prozessmodellierung und des **computational thinking** gegeben.*

1. Einleitung

In der Epidemiologie geht es um die Verbreitung von Krankheitserregern in Populationen [1]. Ist die Ausbreitung global, so spricht man von einer Pandemie, ohne das sich - in wissenschaftlicher Hinsicht - viel ändert. In dieser Abhandlung werden die wesentlichen Eigenschaften einer Epidemie und die zu Grunde liegenden Gesetzmäßigkeiten dargestellt. Dies geht nicht ohne ein gewisses Verständnis für die wissenschaftliche Methodik und führt zudem an eine Denkweise heran, bei der die Verarbeitung und Bereitstellung der Informationen mit Hilfe von Computern erfolgt, das computational thinking.

1.1 Problemstellung

Wir beschränken uns hier auf eine vereinfachte Problemstellung, die jedoch alle wesentlichen Phänomene qualitativ abbilden soll (Parabel):

- die **Ansteckung** erfolgt **über Kontakte** von infizierten Personen mit für die Erreger anfälligen Personen (nicht alle Krankheiten verbreiten sich auf diesem Wege).
- die wesentlichen **Ergebnisse** aus dem **Verlauf der Epidemie** sind
 - die **Dauer** der Epidemie in Vielfachen der charakteristischen Zeitskala (Zeitraum der Infektiosität) t_i ,
 - die **Anzahl der Infizierten** (daraus folgt z.B. die Anzahl der Todesfälle),
 - der Maximalwert der **Neuinfektionsrate** (neu Infizierte pro Zeitraum),
- den Parameter **Reproduktionszahl R** als Maß für Häufigkeit infizierender Kontakte (Kontaktbeschränkungen können R beeinflussen) und
- die Rolle von zufälligen Ereignissen (Kontakten) und der Populationsgröße n_0 .

Reale Epidemien sind komplexer, beinhalten jedoch dieselbe qualitative Phänomenologie wie die hier behandelte Problemstellung. Wichtig ist immer eine wissenschaftlich korrekte Vorgehensweise. Während sich beispielsweise die Öffentlichkeit für Fallzahlen und Todesfälle interessiert, ist für den Wissenschaftler vor allem der zu Grunde liegende Prozess relevant. In immer detaillierteren Zyklen von objektiven Beobachtungen (Messungen) und verfeinerten Modellvorstellungen ergibt sich ein wissenschaftliches Prozessmodell und die Möglichkeit von Modellvorhersagen. Eine echte Modellvorhersage ist daran zu erkennen, das zukünftige Messergebnisse vorhergesagt werden - wie das beispielsweise beim Klimawandel seit 1938 der Fall ist [2,3].

1.2 Divide et impera

Reale Prozesse lassen sich oft nicht sofort in all ihren Details erfassen. Getreu dem Motto *divide et impera* versuchen Wissenschaftler daher zunächst die zu Grunde liegenden Naturgesetze einzeln zu untersuchen und mathematisch zu beschreiben. Durch die nunmehr überall vorhandene Rechenleistung von Computern lassen sich aus den einzelnen Naturgesetzen mittlerweile nicht nur mathematische Beschreibungen im Sinne von Aufgabenstellungen ableiten, sondern diese sind immer häufiger auch einfach lösbar. In diesem Beispiel lernen wir zudem noch eine weitere Möglichkeit kennen, die direkte Simulation durch einen Algorithmus.

Das Motto *divide et impera* spielt auch im Verlauf einer Epidemie eine besondere Rolle: Da in Abwesenheit von Heilmitteln und Impfstoffen eine Epidemie erst zu Ende ist, wenn es keine einzige infektiöse Person mehr gibt, spielt der Zufall und die Größe der Population eine gewisse Rolle. Durch Aufteilen der Population lässt sich somit der Verlauf beeinflussen. Nicht betroffene Teilpopulationen können durch Abschottung geschützt werden.

1.3 Wissenschaft, Wahrheit und Zufall

That is what we meant by science.

That both question and answer are tied up with uncertainty, and that they are painful.

But that there is no way around them.

And that you hide nothing; instead, everything is brought out into the open.

[Peter Høeg]

Der aufmerksame Leser mit Corona-Erfahrung wird es gemerkt haben:

Bei der Problemstellung ist nicht von Fallzahlen und Toten die Rede. Für den Wissenschaftler ist auch das Offensichtliche erst relevant, wenn es zu den wesentlichen Kategorien wissenschaftlichen Denkens passt:

1. Wahrheit

Es wird nach einer Annäherung an die Wahrheit gesucht. Damit muss sich der Wissenschaftler zunächst von Vorurteilen und Eigeninteressen befreien. Zudem wird neutral und emotionslos berichtet. Die Verschleierung von Eigeninteressen ist jedoch nicht einfach zu erkennen.

2. Messungen

Es wird nach Messverfahren gesucht, die relevante Daten liefern. Im Fall der Corona Pandemie sind das an repräsentativen Stichproben durchgeführte Testreihen. Die Messungen können die Fit-Parameter der Modelle liefern oder diese falsifizieren. Messungen ohne eine Betrachtung der Messunsicherheiten sind unwissenschaftlich! Jede Beobachtung, die in mindestens zwei Kategorien einteilbar ist, lässt sich als Messung quantifizieren.

3. Theorie oder Modell

Es wird nach einer vereinfachten Problemstellung gesucht, die zunächst ein grundlegendes Verständnis ermöglicht. Dabei geht es primär um die zu Grunde liegenden Naturgesetze und deren Umsetzung in Form von Modellen und erst sekundär um eine Problemlösung. Die Anwendung der Modelle kann eine Vorhersage oder Problemlösung erlauben, wobei die erforderlichen Simulationen nicht immer auch durchführbar sind. Führt eine Theorie zu Vorhersagen, die im Rahmen der Mess- und Modellierabweichungen nicht mit den Beobachtungen übereinstimmen, so ist das Modell zu verwerfen. Von einer Modellvalidierung wird gesprochen, wenn alle ernsthaften **Falsifizierungsbemühungen** gescheitert sind. Eine Theorie enthält oft Parameter, die indirekt aus Messungen zu bestimmen sind. Dabei kann es sich sowohl um Naturkonstanten handeln als auch um Modell-spezifische Fit-Parameter die noch nicht über Naturgesetze vorhersagbare Zusammenhänge beschreiben. Durch zu viele Fit-Parameter kann es zu einer Überanpassung kommen, wobei ein Modell einen Satz von Messungen wiedergibt ohne das es sich dadurch besonders auszeichnet. Modelle und Naturgesetze müssen einen großen Anwendungsbereich aufweisen.

Eine weitere Kategorie sollte realistischerweise nicht unerwähnt bleiben:

4. Probierkunst:

Als Vorstufe zur Wissenschaft liefert die Probierkunst Erkenntnisse und Werkzeuge, deren Nutzen durch Versuch und Irrtum ermittelt wird. Alle möglichen mathematischen oder algorithmischen Modelle auszuprobieren ist die theoretische Version dieser Vorgehensweise. Wenn sich die Erkenntnisse zu einem Gesamtbild zusammenfügen ist der Übergang zur Wissenschaft möglich.

Alle allgemeinen Kategorien wissenschaftlichen Arbeitens helfen jedoch nicht weiter, wenn die Konzepte für eine qualitative und quantitative Beschreibung einer Problemstellung fehlen oder fehlerhaft sind. Worum geht es also bei einer Epidemie? In der Wirtschaft folgt man der Spur des Geldes, in der Epidemiologie helfen die Spuren der Krankheitserreger!

Relevant ist beispielsweise der Anteil an Infizierten bzw. Betroffenen in der Gesamtbevölkerung, nicht die Fallzahlen im Sinne von erkannten Infektionen. Nur bei einem hundertprozentig zu einer eindeutig identifizierbaren Krankheit führendem Erreger sind beide Größen gleich. Der beste Einstieg ist oft die Definition von quantifizierbaren (mathematischen) Größen und deren Bezug zu den messbaren Größen:

- die Zeit messen wir in Vielfachen des Zeitraumes t_i , über den ein Infizierter die Krankheit weitergeben kann.
- die Intensität der Ausbreitung des Erregers wird über einen Parameter R beschrieben, den wir zunächst als Anzahl der von jedem Infizierten innerhalb von t_i neu angesteckten Personen betrachten.
- alle Maßnahmen zur Eindämmung der Epidemie lassen sich durch Ihren Einfluss auf R quantifizieren. Die Wirksamkeit von Maßnahmen ist somit quantifizierbar und vergleichbar.

Zusätzlich zu den Kausalitäten müssen wir auch mit zufälligen Einflüssen rechnen. Es gilt somit herauszufinden, welche Größen in Wirklichkeit nur über statistische Mittelwerte bestimmbar sind und wie groß die statistischen Schwankungen sind. Dazu vergleichen wir ein deterministisches Modell mit einem stochastischen Modell.

1.4 Occam's Rasiermesser oder das KISS-Prinzip

Warum nehmen wir nicht gleich alle uns bekannten Fakten (z.B. zur Corona-Epidemie) und bauen alles was wir zu wissen glauben zu einem Vorhersagemodell zusammen?

Weil Rom auch nicht an einem Tag erbaut wurde!

Um nicht an der Komplexität einer Aufgabe zu scheitern hilft Vereinfachung, die Ingenieure nennen das KISS-Prinzip. So einfach wie möglich, aber nicht einfacher - hier spricht man auch von Occam's Rasiermesser, was natürlich nicht bedeutet, dass nicht auch schon vor tausenden von Jahren so vorgegangen wurde.

Daher müssen wir zunächst nach dem einfachsten Modell suchen, welches gerade noch alle wesentlichen Phänomene abbildet. Mit diesem Modell können wir (am Computer) Erfahrungen sammeln und die in der Realität wichtigen messbaren Größen identifizieren. In der Regel arbeiten Prozessmodellierer iterativ. Das bedeutet natürlich für den Fall der aktuellen Epidemie, dass Sie auch diese Veröffentlichung nicht in die Lage versetzen wird, den konkreten weiteren Verlauf genau vorherzusagen. Sie werden jedoch lernen, das Mögliche vom Unmöglichen zu unterscheiden, die Kausalitäten von den Zufallsereignissen zu unterscheiden und die Konsequenzen von politischen Entscheidungen besser einzuschätzen.

1.5 Entscheidungsproblem oder Regelungsproblem

Der Verlauf einer Epidemie kann beeinflusst werden. Dazu ist die Reproduktionszahl R zu beeinflussen. Jede Beeinflussungsmaßnahme B_i ist mit Aufwand verbunden und in ihrer Wirkung begrenzt. Das Ergebnis aller Maßnahmen kann in dem dadurch erreichten R zusammengefasst werden. Es ist offensichtlich, dass ein Wert von $R < 1$ zu einem Absinken und $R > 1$ zu einem Anstieg der Infektionszahlen führt. Für $R \gg 1$ sind schnell fast alle Mitglieder der Population betroffen, für $R \ll 1$ ist die Epidemie schnell wieder vorbei und für $R \sim 1$ dauert es sehr lange, bis alle Mitglieder der Population betroffen sind.

Außerhalb der Aufgabenstellung dieser Arbeit liegt die Bestimmung der Beeinflussungsmöglichkeiten B_i und deren kumulativer Wirkung $R(B_1, \dots, B_i, \dots, B_n)$. Nun lassen sich den B_i Kosten $K_{i,j}$ verschiedener Kategorien j zuordnen. Diese Kosten sind nicht in beliebiger Höhe tragbar und es ist vorstellbar, eine oder mehrere Gesamtkostenfunktionen aufzustellen und optimieren zu wollen. Als Ergebnis würde sich dann eine optimale Beeinflussung der Epidemie ergeben, in der Regel ein Pareto-Optimum. Liegt dieses Optimum immer in einer Ecke des Parameterraumes der Beeinflussungsmöglichkeiten, so ist nur die Entscheidung sich in diese Ecke zu begeben erforderlich. Andernfalls wird ein Modellprädiktiver Regler benötigt um die Beeinflussung optimal zu gestalten. Hinweise auf die Natur einer einfachen Problemlösung kann hingegen auch schon ein stark vereinfachtes Modell geben.

1.6 Modellierkonzepte

Ein Modell ist ein vereinfachtes Abbild der Wirklichkeit. Modelle können unser unstillbares Verlangen nach Gründen (Kausalitäten) befriedigen oder sie können verwendet werden, um die Zukunft vorherzusagen. Um das Konzept und die Rolle von Modellvorhersagen zu verstehen helfen Beispiele:

1. In der **Raumfahrt** ist die Vorausberechnung von Flugbahnen unverzichtbar und es ist in der Regel auch eine Steuerung unter Verwendung der Modelle erforderlich. Es gilt das Prinzip **Compute 1st**.
2. Beim **Glücksspiel** gibt es keine Vorhersage eines konkreten Ereignisses, die Wahrscheinlichkeiten der möglichen Ereignisse lassen sich jedoch berechnen.
3. Bei der **Wettervorhersage** ist es in jahrzehntelanger Arbeit gelungen, das zukünftige Wetter im Sinne einer Wahrscheinlichkeitsverteilung für einen begrenzten Zeitraum von wenigen Tagen vorherzusagen [4]. Dazu sind nicht nur aufwendige Modelle, sondern auch ebenso aufwendige Messungen des aktuellen weltweiten Wetters erforderlich. Die Trefferwahrscheinlichkeit einer Wettervorhersage hängt von folgenden Faktoren ab:
 - Dem lokalen Klima, welches nichts anderes als die Wahrscheinlichkeitsverteilung für das lokale Wetter darstellt.
 - Der Genauigkeit mit der das momentane globale Wetter gemessen werden kann, denn diese Messwerte werden als Anfangswerte für die Modellrechnungen benötigt.
 - Der Qualität der Modelle, weitgehend bestimmt von
 - der Richtigkeit der mathematisch - algorithmischen Beschreibung,
 - der rechnerischen Genauigkeit (Auflösung),
 - der Abbildung der verbleibenden Stochastik und

- der benötigten Datenbasis.

4. Bei der **Klimamodellierung** potenzieren sich die Schwierigkeiten der Wettervorhersage, es handelt sich um eines der anspruchsvollsten wissenschaftlichen Teilgebiete:

- Grundlegende Effekte sind qualitativ unstrittig, ihre genaue Quantifizierung ist jedoch oft eine Herkulesaufgabe (More is different).
- Vorhersagen sind unverzichtbar: In bewohnten Gebieten lässt sich das heutige Wetter überleben, das wissen wir aus Erfahrung. In welchen Gebieten ist das auch in Zukunft noch für wie viele Menschen möglich? Gibt es einen "normalen" Klimawandel an den sich die Biosphäre anpassen kann oder handelt es sich beim Anthropozän um eine Klimakatastrophe die erst lange nach dem Aussterben der Verursacher wieder in ein stabiles Klima (oder den galoppierenden Treibhauseffekt) mündet?
- Messungen zur Einschätzung der Qualität der Vorhersagen sind schwierig: Klimawandel fand bisher in der Regel über lange Zeiträume statt und wird erst seit einigen Jahrzehnten wissenschaftlich untersucht.
- Im Gegensatz zu einer (beeinflussbaren) Pandemie sind die Zeitskalen inkompatibel zu den menschlichen Planungshorizonten, es gibt Kippelemente.

Nun könnte die Befürchtung aufkommen, das ein epidemisches Modell ähnlich komplex wie ein Klimamodell werden könnte. Nun dann wäre es nicht als Beispiel für diese Arbeit in Frage gekommen. Zudem gibt es auch in der Erdsystemwissenschaft Parabeln [5], die uns grundlegende Effekte einfach verstehen lassen.

1.7 Aufbau dieser Arbeit

Im folgenden Abschnitt 2 wird ein Simulationsprogramm vorgestellt, welches den Verlauf einer Epidemie unter der Voraussetzung zufälliger Kontakte innerhalb der Population simuliert. Die von diesem Programm gelieferten Mittelwerte werden in Abschnitt 4 mit denen des deterministischen SIR-Modells aus Abschnitt 3 verglichen. Es folgt eine detaillierte Diskussion der Modellparameter, insbesondere der Reproduktionszahl R in Abschnitt 5. So gerüstet lassen sich die beiden Modelle dazu verwenden, die den Verlauf einer Epidemie kennzeichnenden Größen zu definieren und zu berechnen. Dies geschieht in Abschnitt 6, in dem dann auch die prinzipiellen Möglichkeiten einer Beeinflussung einer Epidemie diskutiert werden.

2. Das Simulationsmodell

Krankheitserreger werden durch Kontakte übertragen und diese Kontakte sind nicht immer vorher-sagbar. Wir suchen daher zunächst nach einem Modell, welches die Zufälligkeit der Kontakte berücksichtigt. Wir versuchen es zunächst mit einem einfachen Simulationsalgorithmus.

2.1 Modellannahmen

Um ein Modell des Verlaufes einer Epidemie zu bekommen, benötigen wir quantifizierbare Größen die das System in seinen wesentlichen Aspekten möglichst einfach beschreiben:

1. Wir gehen von einer zeitlich konstanten Bevölkerungszahl n_0 aus. Auch eine tödliche Seuche kann so beschrieben werden, da es für die Ausbreitung der Epidemie keine Rolle spielt, in welchem Zustand sich eine nicht mehr infizierbare und nicht mehr infektiöse Person befindet.
2. Jede Person kann drei Zustände annehmen:
infizierbar ($S=Suceptible$), infektiös ($I=Infected$) und unbeteiligt ($R=Removed$).
3. Unter den anfänglich n_0 Personen befinden sich n_{I0} infektiöse (I) und n_{R0} unbeteiligte (R). Der Rest ist infizierbar (S).

Nun benötigen wir eine Handlungsvorschrift für den Ablauf unserer Simulation, einen Algorithmus.

2.2 Der Simulationsalgorithmus

Das im nächsten Abschnitt publizierte Computerprogramm **SimulateEpidemic** besteht aus verschiedenen Teilen, von denen wir hier nur den Kern, d.h. den Simulationsalgorithmus besprechen wollen:

1. Festlegen der Simulationsparameter: im Wesentlichen n_0 und R .
2. Initialisieren der Simulationsdaten:
 n_{I0} der n_0 Personen auswürfeln und als infiziert kennzeichnen.
3. Schrittweise Durchführung der Simulation:
Die Daten des Zeitschrittes $t=1$ wurden in der Initialisierung festgelegt.
Nun folgen die weiteren Zeitschritte, bis der maximale Simulationszeitraum erreicht ist:
 - 3.1. Aus den im vorherigen Schritt als infiziert Gekennzeichneten werden Betroffene ($Removed$), die nicht noch einmal neu infiziert werden können (Annahme!).
 - 3.2. In Abhängigkeit von der Anzahl der Infektiösen werden neue Infektionen durch zufällige Kontakte ausgewürfelt.
 - 3.3. Es wird etwas Buchhaltung betrieben um den Verlauf der interessanten Größen zu ermitteln.

Diese Beschreibung in Umgangssprache liefert bestenfalls einen Eindruck der Abläufe. Wirklich nachvollziehen lässt sich ein Algorithmus nur in seiner natürlichen Formulierung in einer Programmiersprache (Computational Language). Dazu finden Sie im Abschnitt 2.3 das komplette Programm. Mit Mathematica (ist zum Beispiel auf einem Raspberry Pi schon vorinstalliert) und Kenntnissen in der Wolfram Language können Sie das Programm selbst abändern.

2.3 Das Simulationsprogramm (Wolfram Language)

In[83]:=

```
(* (c) 2020 jens@wendelstorf.de *)
SimulateEpidemic::usage =
  "SimulateEpidemic[{parameter->value,...}]
  is an Agent Based Model (random contacts) of an epidemic (t=1..tMax)
  using the primary parameters n0, nI0, nR0, qC and R. Additionally, interventions can be described by
  the parameters xdImax, xdImin, RI and nTestsMax. It returns the population fractions of S=Susceptible,
  I=Infected, dI=newly Infected, R=Removed and Q=Quarantined as {[0..1]} vectors of length tMax.";

SimulateEpidemic["DefaultArguments"]={
  "n0" -> 1000, "nI0" -> 10, "nR0" -> 0,
  "qC" -> 0., "nTestsMax" -> 10000000, (* NYI *)
  "R" -> 4., "tMax"->10, (* NYI: tMax->Automatic *)
  "PDF" -> "None", (* "Poisson", "Geometric" *)
  "xdImax" -> 1., "xdImin" -> 0., "RI" -> 4. (* no intervention *)
};

SimulateEpidemic[input:{__Rule}] := Module[
{
  cI, (* index list of contacts *)
  i, iSum, iI, (* # of infections *)
  fn, fR, (* the actual reproduction rate *)
  fracQ,
  nt, (* # of time steps *)
  new, newInteger, (* infections from reproduction rate *)
  nI, nS, nDI, nR, nQ,
  n0, nR0, nI0, Ract,
  ps, (* state: 0=Susceptible, 1=Infected, 2=Removed *)
  r, t = 1,
  x, xR, xRI
},
  x = <|Join[SimulateEpidemic["DefaultArguments"],input]|>;
  Which[(* # of infected may be random: *)
    x["PDF"]=="Poisson",
      fn[r_?NumericQ]:=First[RandomVariate[PoissonDistribution[Max[r,10^-8]],1]];,
    x["PDF"]=="Geometric",
      fn[r_?NumericQ]:=First[RandomVariate[GeometricDistribution[1/(1.+r)],1]];,
    True,
      fn[r_?NumericQ]:=r;
  ];
  fracQ = Min[Max[1. - x["qC"], 0.], 1.];
  n0 = x["n0"]; nR0 = x["nR0"]; nI0 = x["nI0"];
  (* initialize state and results: *)
  nS = nI = nR = nQ = Table[0., {t, x["tMax"]};];
  Ract = Table[1., {t, x["tMax"]};];
  ps = Table[0, n0];(* the actual state of the population *)
  If[nR0 > 0, ps[[cI=RandomSample[Range[n0], Min[nR0, n0]]]] = 1];
  (* ToDo: Sample should not be in cI! *)
  ps[[RandomSample[Range[n0], Min[nI0, n0]]]] = 2;
  i = nI0; iI = i; (* nobody is initially identified *)
  iSum = i; r = nR0; fR = xR = x["R"];
  xRI = If[x["RI"]==Undefined, xR, x["RI"]];
  nS[[t]] = N[(n0 - iSum - r)/n0];
  nI[[t]] = N[i/n0];
  nR[[t]] = N[r/n0];
  (* simulate max. nt days *)
  While[t < x["tMax"],
    t++;
    fR=If[fR==xR && (i/n0 > x["xdImax"]),
      xRI, (* else *)
      If[ fR==xRI && (i/n0 < x["xdImin"]),
        xR, fR]
```



```

];
r += i; (* old infected => removed *)
new = fn[fR*iI]; (* new infections *)
newInteger = Floor[new]; (* # of new possible infections: *)
If[RandomReal[] < new - newInteger, newInteger++];
i = 0; iI = 0; (* i counts new infections, iI the identified *)
cI = RandomInteger[{1, n0}, Min[newInteger, n0]]; (* random contacts *)
(* random contacts without duplicates:
cI = RandomSample[Range[n0], Min[newInteger, n0]]; *)
If[ ps[ [#]] == 0,
  ps[ [#]] = 1;
  (* ToDo: Quarantine contacts of the infected! *)
  If[ (RandomReal[] < fracQ) ||
    ((i-iI) > x["nTestsMax"]), iI++]; (* infectious *)
  i++;
] & /@ cI; (* infect the susceptible *)
iSum += i; (* count cases *)
(* store in numerical result vectors: *)
nS[[t]] = N[(n0 - i - r)/n0];
nI[[t]] = N[i/n0];
nR[[t]] = N[r/n0];
nQ[[t]] = N[(i - iI)/n0];
Ract[[t]] = N[fR/x["R"]];
]; (* return result: *)
{"S" -> nS, "I" -> nI, "R" -> nR, "Ract" -> Ract}
]; (*END:SimulateEpidemic*)

SimulateEpidemicVisualization::usage = "
SimulateEpidemicVisualization[{parameter->value,...}]
  visualizes the results of Agent Based Model (random contacts) of an epidemic (t=1..tMax)
  using the primary parameters n0, nI0, nR0, qC and R. Additionally,
  interventions can be described by the parameters xDImin, xDImin, RI and nTestsMax.";

SimulateEpidemicVisualization[input:{___Rule}]:=Module[
  {pTitle, r, x},
  x = <|Join[SimulateEpidemic["DefaultArguments"],input]|>;
  r = <|SimulateEpidemic[input]|>;
  pTitle = "n0="<>ToString[x["n0"]]<>
    ", nI0="<>ToString[x["nI0"]]<>, nR0="<>ToString[x["nR0"]]<>
    ", R="<>ToString[x["R"]]<>, RI="<>ToString[x["RI"]]<>
    ", xDImin="<>ToString[FortranForm[x["xDImin"]]]<>
    ", xDImin="<>ToString[FortranForm[x["xDImin"]]]<>;
  ListPlot[{r["S"], r["I"], r["R"], r["Ract"]},
    Frame->True, PlotRange -> All,
    FrameLabel -> {
      {"population fraction [0..1]",
        Style["SimulateEpidemic © 2020 by jens@wendelstorf.de",Tiny]},
      {"xt", Style[pTitle,Tiny]}}},
    PlotLegends -> {"S", "I", "R", "Ract/R0"}, Joined->True]
]; (*END:SimulateEpidemicVisualization*)
(* for Manipulate: *)
ClearAll[sev];sev[i_]:=sev[i]=SimulateEpidemicVisualization[i];

SimulateEpidemicStatistics::usage = "
SimulateEpidemicStatistics[nRuns_Integer, input:{___Rule}]
  gives a Dataset with the Min, Max, Mean, Median,
  SD values for the t=1..tMax and S, I, R, dI and Q";

SimulateEpidemicStatistics[nRuns_Integer, input:{___Rule}]:=
  SimulateEpidemicStatisticsComputation[
    Table[<|SimulateEpidemic[input]|>, {i,nRuns}]];

SimulateEpidemicStatisticsComputation[r_List]:=Module[
  {nt,keys,nKeys,nSamples,rStat},
  nt=Length[r[[1,1]]];

```

```

keys=r[[1]]//Keys;
nKeys=Length[keys];
nSamples=Length[r];
rStat=Table[Block[{d},
  d = Table[r[[i]][keys[[k]]][[t]],{i,nSamples}];
  {Max[d], Min[d], Mean[d], Median[d], StandardDeviation[d]}],
  {t, nt}, {k, nKeys}
];(* t, key, value *)
Dataset[<|Table[keys[[k]] → <|
  "Max" → rStat[[All, k, 1]], "Min" → rStat[[All,k,2]],
  "Mean" → rStat[[All,k,3]], "Median" → rStat[[All,k,4]],
  "SD" → rStat[[All,k,5]] |>, {k,nKeys}]]>]
];(*END:SimulateEpidemicStatisticsComputation*)

SimulateEpidemicOutcome::usage="
SimulateEpidemicOutcome[nRuns_Integer, input:{___Rule}]
  gives the main results of an epidemic performing nRuns simulations.";

SimulateEpidemicOutcome[nRuns_Integer, input:{___Rule}]:=Module[
  {ds, tMax, x},
  x = <|Join[SimulateEpidemic["DefaultArguments"],input]|>;
  tMax = x["tMax"];
  (* ToDo tMax Automatic und Berechnung der Dauer der Epidemie *)
  ds = SimulateEpidemicStatistics[nRuns,input];
  Dataset[{<|Join[{ "inputs"→<|"n0"→x["n0"], "tMax"→ tMax|>,
    (# → <|"SResult" → ds["S",#,tMax], "IMax" → Max[ds["I",#,All]],
    "RResult" → ds["R",#,tMax], "QMax" → Max[ds["Q",#,All]]|>
    & /@ {"Min", "Max", "Mean", "Median", "SD"}|>}]
];(* END:SimulateEpidemicOutcome *)

SimulateEpidemicStatisticsVisualization::usage="
SimulateEpidemicStatisticsVisualization[nRuns_Integer, input:{___Rule}]
  visualizes the Statistics of nRuns simulations runs.";

SimulateEpidemicStatisticsVisualization[nRuns_Integer, input:{___Rule}]:=Module[
  {ds, r, plot, legends, pTitle, tabs, x},
  r = Table[<|SimulateEpidemic[input]|>, {i, nRuns}];
  ds = SimulateEpidemicStatisticsComputation[r];
  tabs = r // First // Keys // Normal;
  legends = ds // First // Keys // Normal;
  x = <|Join[SimulateEpidemic["DefaultArguments"],input]|>;
  pTitle = "n0="<>ToString[x["n0"]]<>
    ", nI0="<>ToString[x["nI0"]]<>, nR0="<>ToString[x["nR0"]]<>
    ", R="<>ToString[x["R"]]<>, RI="<>ToString[x["RI"]]<>
    ", xDImin="<>ToString[x["xDImin"]]<>, xDImax="<>ToString[x["xDImax"]];
  plot[key_] := ListPlot[ds[key,#] & /@ (ds//First//Keys),
    Joined → True, PlotLegends → legends, PlotRange → Full,
    Frame → True, FrameLabel → {
      {"population fraction [0..1]",
        Style["SimulateEpidemic © 2020 by jens@wendelstorf.de",Tiny]},
      {"xt", Style[pTitle,Tiny]}}];
  TabView[#:plot[#] & /@ tabs]
];(*END:SimulateEpidemicStatisticsVisualization*)

ClearAll[sesv];(* function with memory for Manipulate *)
sesv[nRuns_Integer, input:{___Rule}]:=sesv[nRuns, input]=
  SimulateEpidemicStatisticsVisualization[nRuns, input];

SimulateEpidemicParameterVariation::usage="
SimulateEpidemicParameterVariation[n, input, p]
  visualizes the parameter dependency for n simulations using the parameter iterator p";

SimulateEpidemicParameterVariation[nRuns_Integer, input:{___Rule}, p_List]:=Module[
  {i, iterator, pData, rData, sData, n, x},
  iterator = {x, p[[2]], p[[3]], p[[4]]};
  sData = Table[SimulateEpidemicOutcome[100, Join[input, {First[p] → x}]], Evaluate[iterator]];

```

```

rData = Table[x, Evaluate[iterator]]; n=Length[rData];
pData[r_String, s_String] := Transpose[{rData, Table[sData[[i]][1, s, r], {i,n}]}];
TabView[{
  "R∞" → ListPlot[{
    pData["RResult", "Min"], pData["RResult", "Mean"], pData["RResult", "Max"]},
    Joined→True, PlotLegends→{"Min", "Mean", "Max"},
    PlotRange→Full, Frame→True, FrameLabel→{First[p], "R∞"},
  "Imax" → ListPlot[{
    pData["IMax", "Min"], pData["IMax", "Mean"], pData["IMax", "Max"]},
    Joined→True, PlotRange→Full, PlotLegends→{"Min", "Mean", "Max"},
    Frame→True, FrameLabel→{First[p], "Imax"}}
  ]}
]; (*SimulateEpidemicParameterVariation*)

```

2.4 Parameterabhängigkeiten

Für kleine Populationen wird der Verlauf einer Epidemie auch vom Zufall bestimmt. Dabei können nicht nur die Kontakte als Zufallsereignisse betrachtet werden sondern auch deren Anzahl. Um Letzteres zu simulieren haben wir drei Möglichkeiten in das Simulationsprogramm eingebaut:

- "PDF" → "None":
Jede Infektion führt zu genau **R** weiteren Infektionen.
- "PDF" → "Poisson":
Jede Infektion führt im Mittel zu **R** weiteren Infektionen, wobei die jeweilige Anzahl einer Poisson-Verteilung folgt.
- "PDF" → "Geometric":
Jede Infektion führt im Mittel zu **R** weiteren Infektionen, wobei die jeweilige Anzahl einer geometrischen Verteilungsfunktion folgt. Diese streut derart stark, dass eine Epidemie in einer kleinen Population durchaus zufällig verschwinden oder auch bei kleinerem **R** katastrophal enden kann.

Wir können nun eine theoretisch bekannte Parameterabhängigkeit simulieren:

Je nach Reproduktionsfaktor ist nur ein Teil oder fast die gesamte Population von der Epidemie betroffen. Diese Abhängigkeit $R_{\infty}(R)$ lässt sich mathematisch berechnen und sollte sich auch aus der Simulation ergeben. Zusätzlich erlaubt die Simulation auch noch eine Einschätzung der Statistik dieses Zusammenhanges, die ja für kleine Populationsgrößen durchaus eine Rolle spielen kann. Nehmen wir also eine Population von 200 an und starten mit 2 Infizierten. Die im Epidemieverlauf auftretende Maximalzahl von Infizierten I_{\max} ist ein weiteres Ergebnis. Die 56 Simulationen können Ihren PC durchaus 1 Minute beschäftigen!

```

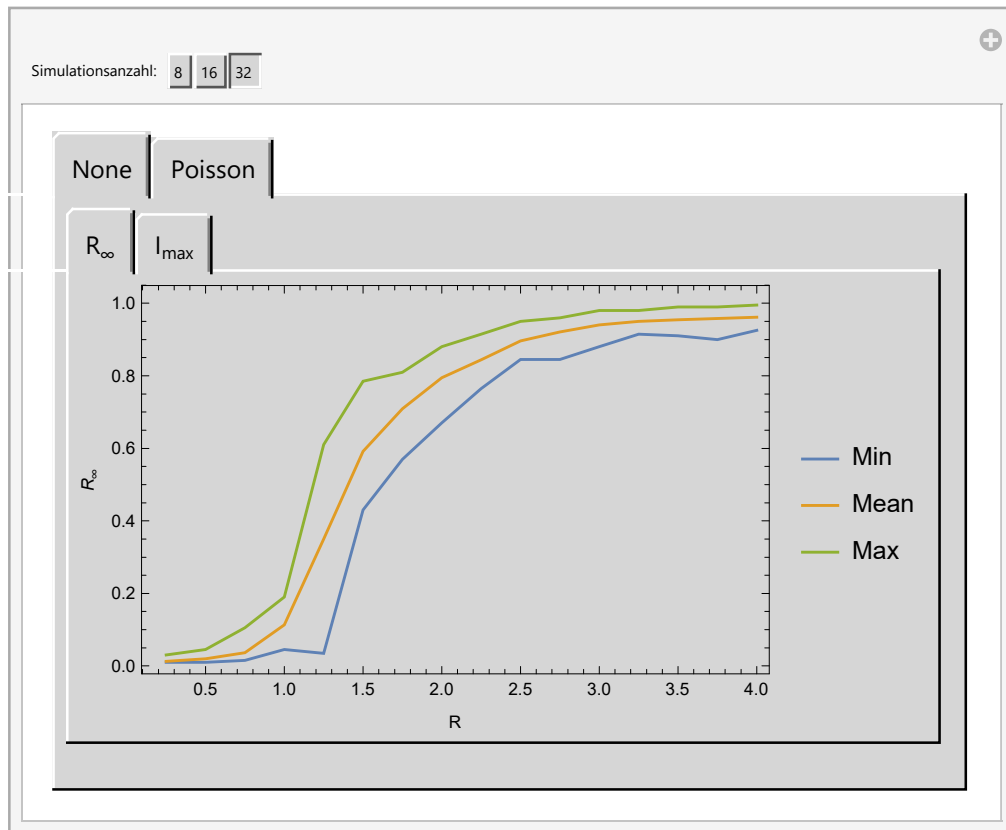
In[118]:= inp = {"n0" → 200, "nI0" → 2, "tMax" → 20, "xdImax" → 1., "xdImin" → 0.};
ClearAll[fSP]; nList = {8, 16, 32};

fSP[n_?NumericQ] := fSP[n] = TabView[...]; Table[fSP[n], {n, nList}];

Manipulate[Evaluate[fSP[n]], {{n, Last[nList], "Simulationsanzahl:"}, Evaluate[nList]}, SaveDefinitions → True]

```

Out[121]=



2.5 Die interaktive Simulation

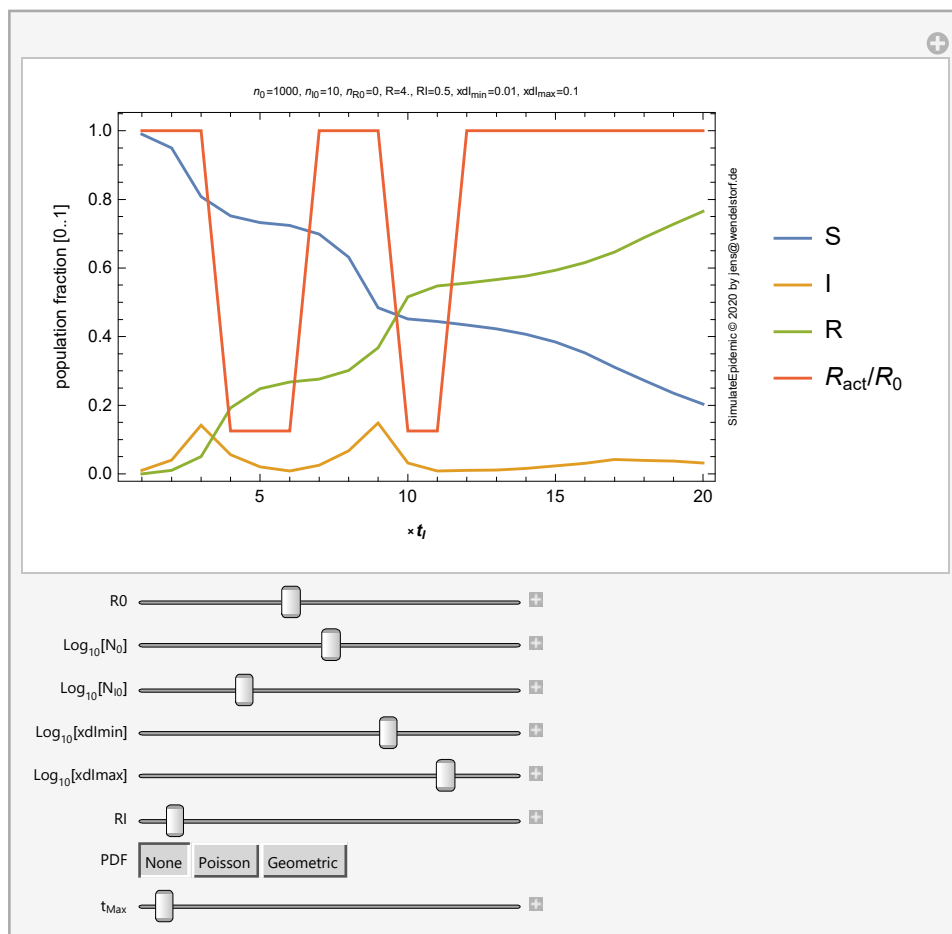
2.6.1 Simulation einer einzelnen Epidemie

Mit dem folgenden Werkzeug können Sie den Verlauf einer einzelnen Epidemie simulieren. Die Vorgaben der Parameterwerte ergeben eine Epidemie mit (zu früh beendeter) Intervention. Je kleiner die Population und die Zahl der anfänglich Infizierten, um so stärker wirkt sich der Zufall aus:

Sie bekommen für ein und dieselben Parameterwerte unterschiedliche Verläufe.


In[117]: Manipulate[...] +

Out[117]:

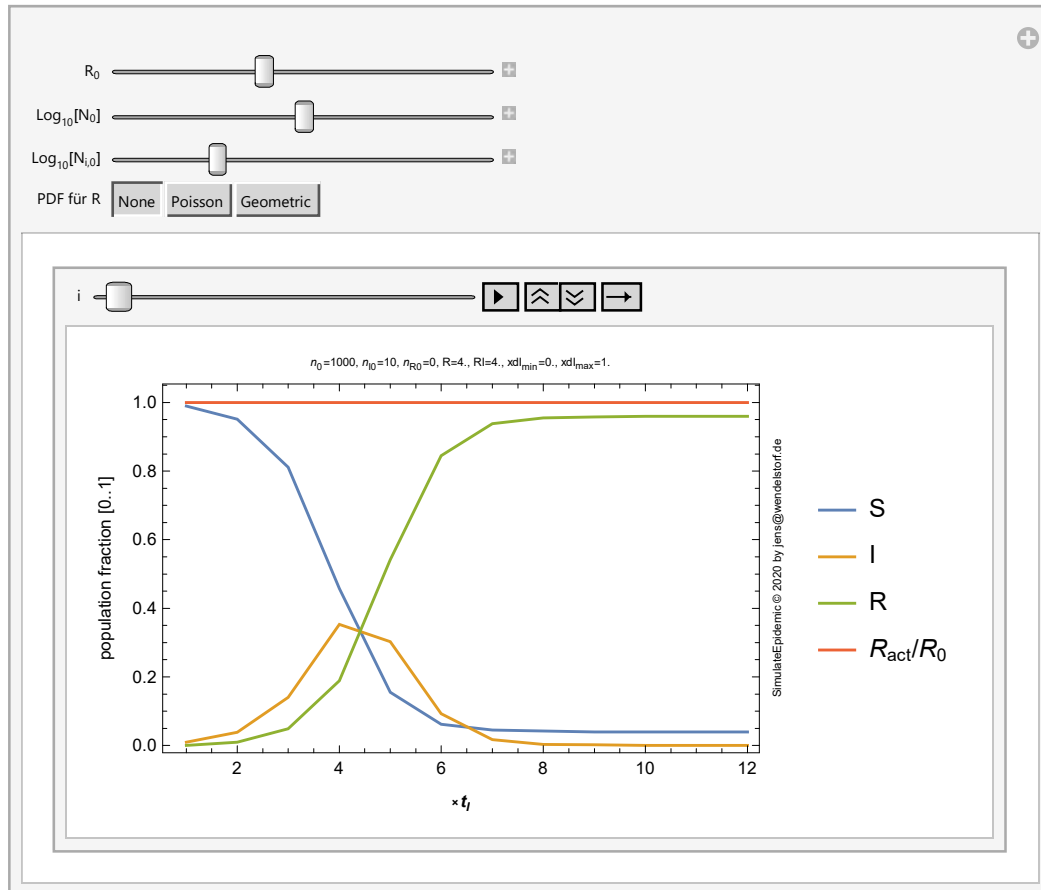


2.6.2 Simulation einer Epidemie (Animiert)

Eine nützliche Technik zur Visualisierung besteht in der Erzeugung interaktiver Videos.

Klicken Sie innerhalb der Grafik auf Play  und Sie erhalten einen Eindruck von der Rolle des Zufalls beim Verlauf der Epidemie:

In[77]:= Manipulate[...]



Out[77]=

2.6.3 Die Rolle des Zufalls (Statistik)

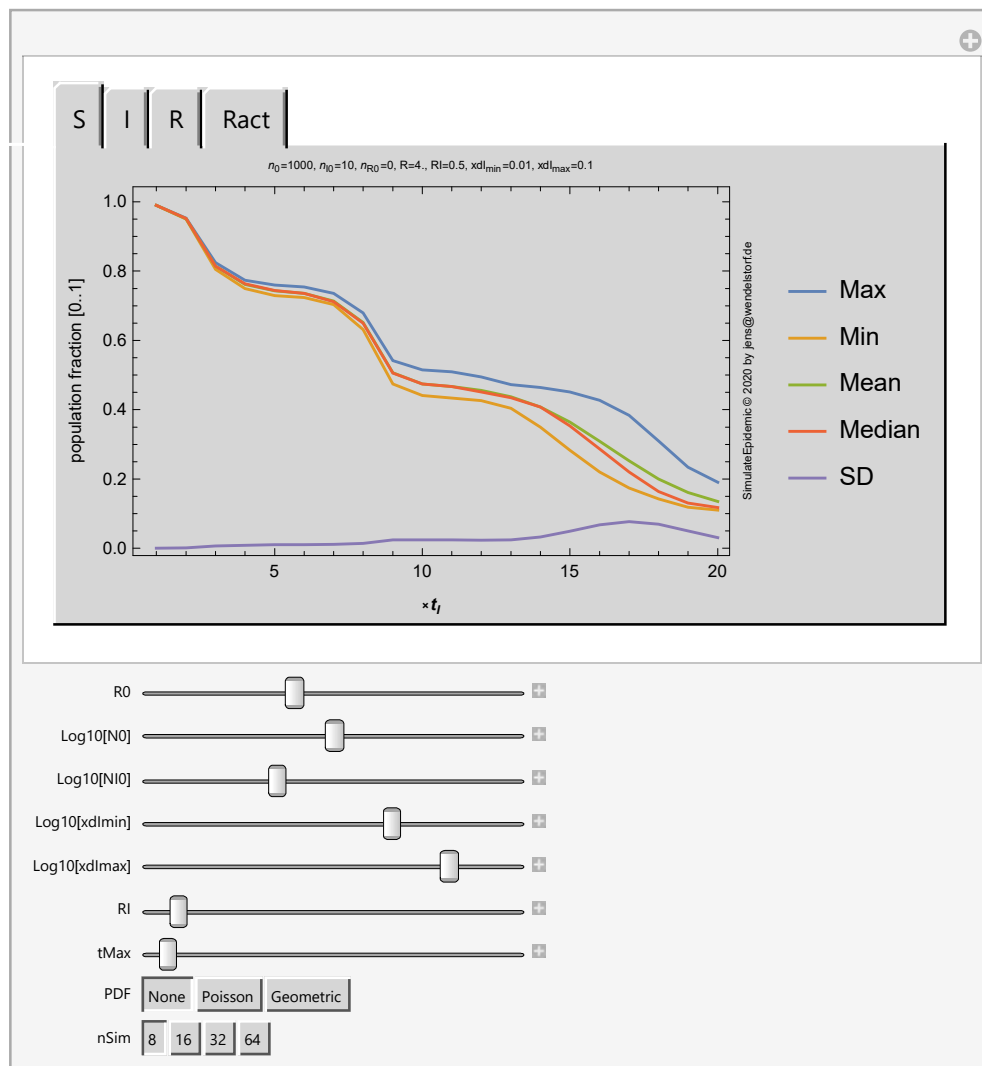
Mit dem folgenden Werkzeug können Sie die Stochastik genauer studieren. Dazu erhalten Sie eine statistische Auswertung von bis zu 64 Simulationen.

Vorsicht: Seien Sie geduldig, sonst kommt Ihr Computer mit dem Rechnen nicht nach.

In[78]:=

Manipulate[...]

Out[78]:=



2.6 Diskussion des Simulationsmodells

Gerade für Prozesse, bei denen die Rolle von Zufallsereignissen nicht zu vernachlässigen ist, liefern Simulationsmodelle detaillierte Ergebnisse. Die Vorteile liegen in den Möglichkeiten beliebige Details zu berücksichtigen. Darin liegt jedoch gleichzeitig auch ein Nachteil, da es schwer ist realistische Simulationsmodelle von solchen zu unterscheiden, bei denen lediglich die Parameter so gewählt wurden, dass sich ein gewünschtes Ergebnis ergibt. Es ist somit darauf zu achten, dass möglichst viele Parameter unabhängig vom Prozess gemessen werden können, da nur bei einer kleinen Anzahl von verbleibenden Fit-Parametern eine Chance besteht, diese aus Beobachtungen abzuleiten. Um quantitative Vorhersagen zu erzielen ist oft eine langwierige und aufwendige Entwicklungsarbeit notwendig, im Laufe derer Beobachtungsmöglichkeiten und Modelle immer weiter verfeinert werden.

Für unser Problem, den Verlauf einer Epidemie, gibt es gesicherte wissenschaftliche Erkenntnisse über die auftretenden qualitativen Phänomene und es gibt mathematische Modelle mit denen die wesentlichen Phänomene von der Stochastik (dem *zufälligen Wackeln*) unterschieden werden können. Diesem Thema wenden wir uns im nächsten Abschnitt zu und lernen ein Beispiel für einen nützlichen und sehr oft anwendbaren Typ von mathematischen Modellen kennen.

3. Das SIR Modell mit Differentialgleichungen

In Abschnitt 2 haben wir ein stochastisches **Simulationsmodell** kennengelernt. In der Fachliteratur wird der Verlauf einer Epidemie oft mit Hilfe von Abkömmlingen des SIR-Modells erklärt. Dieser Ansatz verwendet ein **mathematisches Modell**, dessen Eigenschaften auch ohne Computer untersucht werden können [1]. Dabei werden Differentialgleichungen für den zeitlichen Verlauf aufgestellt und untersucht. Der Zufall, der den Verlauf einer Epidemie anfangs und am Ende durchaus beeinflussen kann, bleibt unberücksichtigt. Im Mittel liefert das Simulationsmodell aus dem vorherigen Abschnitt ähnliche Ergebnisse wie das folgende SIR-Modell, zusätzlich liefert das Simulationsmodell auch Informationen über die statistische Variabilität der Kennwerte des Epidemieverlaufes.

3.1 Das einfache **SIR** - Modell

Voraussetzung für das Verständnis dieses mathematischen Modells ist das Konzept der gewöhnlichen Differentialgleichungen, wobei hier gleich auf die DAE-Systeme verwiesen wird, deren Anwendungsbereich noch deutlich größer ist, insbesondere in den Ingenieurwissenschaften.

Das **SIR**-Modell [1,6,Wikipedia,WolframMathWorld] ist nach den bereits eingeführten Gruppen benannt, in die die Population eingeteilt wird:

- **S**=*Susceptible*, die anfangs oft fast die gesamte Population umfassende Gruppe der Infizierbaren,
- **I**=*Infected*, die Infizierten in dem Zeitraum, in dem Sie die Infektion weitergeben können.
- **R**=*Removed*, die für das Infektionsgeschehen nicht mehr relevante Gruppe.

Zusammen mit der Gesamtzahl $N=S+I+R$ ergeben sich zeitabhängige Anzahlen, die über Ratengleichungen miteinander verknüpft werden ($f'[t]$ symbolisiert die Ableitung $\frac{df}{dt}$)

$$\begin{aligned} S'[t] &= -\alpha S[t] \times I[t] \\ I'[t] &= \alpha S[t] \times I[t] - I[t] \\ R'[t] &= I[t] \end{aligned}$$

und für die Anfangsbedingungen

$$\begin{aligned} S[t = 1] &= S_0 \\ I[t = 1] &= I_0 \\ R[t = 1] &= R_0 \end{aligned}$$

festgelegt werden müssen (die eckigen Klammern werden in der Wolfram Language für die Argumente einer Funktion verwendet). Das hier verwendete SIR Modell ist dasjenige mit der kleinsten Zahl von Parametern, welches mit der Simulation vergleichbare Ergebnisse liefert (konstantes N , $\gamma=1$).

3.1.1 Der **SIRModel**-Quellcode (Wolfram Language)

```

In[100]:= (* (c) 2020 jens@wendelstorf.de *)
SIRModel::usage =
  "SIRModel[{parameter->value,...}]
  is an SIR Model of an epidemic (t=1..tMax)
  using the primary parameters n0, nI0, nR0 and R.
  It returns the population fractions of S=Susceptible,
  I=Infected and R=Removed as {[0..1]} vectors of length tMax.";

SIRModel["DefaultArguments"]={
  "n0" -> 1000, "nI0" -> 10, "nR0" -> 0,
  "dtOut" -> 1., (* for demonstrating the continuity of the model *)
  "R" -> 4., "tMax"->10.
};
(* ToDo: tMax->Automatic *)

SIRModel[input:{__Rule}] := Module[
  { alpha, dtOut, eqns,
    fS, fI, fR, fracQ, fS0, fI0, fR0,
    fStepM, fStepP, fNPIOon, fNPIOff,
    tNPI,
    n0, nR0, nI0,
    s, t, tMax, vars, x,
    yS, yI, yR, yQ, yalpha
  },
  x = <|Join[SIRModel["DefaultArguments"],input]|>;
  tMax = x["tMax"]; tNPI=2 tMax;
  dtOut = x["dtOut"]; alpha = x["R"];
  n0 = x["n0"]; nR0 = x["nR0"]; nI0 = x["nI0"];
  fI0 = N[nI0/n0]; fR0 = N[nR0/n0]; fS0 = 1. - fI0 - fR0;
  (* compared to the original model, we set beta=1,
    thus the time is scaled by the infectious time of a person *)
  eqns = { (* differential equations for fS, fI, fR: *)
    fS'[t]== -alpha fS[t]×fI[t],
    fI'[t]== alpha fS[t]×fI[t] - fI[t],
    fR'[t]== fI[t], (* check: R+I+S=n0 *)
    (* initial conditions for fS, fI, fR: *)
    fS[1.]== fS0, fI[1.]== fI0, fR[1.]== fR0;
  vars = {fS, fI, fR};
  s = First[NDSolve[eqns, vars, {t, 1., tMax},
    StartingStepSize->0.01, MaxStepSize->0.1]];
  tNPI = 2 tMax;
  yS = Table[Evaluate[fS[t] /. s], {t, 1., tMax, dtOut}];
  yI = Table[Evaluate[fI[t] /. s], {t, 1., tMax, dtOut}];
  yR = Table[Evaluate[fR[t] /. s], {t, 1., tMax, dtOut}];
  {"S" -> yS, "I" -> yI, "R" -> yR}
];(*END:SIRModel*)

SIRModelVisualization[input:{__Rule}] := Module[
  {pTitle, r, x},
  x = <|Join[SIRModel["DefaultArguments"],input]|>;
  r = <|SIRModel[input]|>;
  pTitle = "n0="<>ToString[x["n0"]]<>
    ", nI0="<>ToString[x["nI0"]]<>
    ", nR0="<>ToString[x["nR0"]]<>
    ", R="<>ToString[x["R"]];
  TabView[{
    "Linear scale" -> ListPlot[{r["S"], r["I"], r["R"]},
      Frame -> True, PlotRange -> All,
      FrameLabel -> {
        {"population fraction [0..1]",
          Style["SIRModel © 2020 by jens@wendelstorf.de", Tiny]},

```

```

    {"xt", Style[pTitle, Tiny]}},
    PlotLegends → {"S", "I", "R"}, Joined → True],
    "Logarithmic scale" → ListLogPlot[{r["S"], r["I"], r["R"]},
    Frame → True, PlotRange → {x["nI0"]/x["n0"]/10, 1.},
    FrameLabel → {
        {"population fraction [0..1]",
        Style["SIRModel © 2020 by jens@wendelstorf.de", Tiny]}},
        {"xt", Style[pTitle, Tiny]}},
    PlotLegends → {"S", "I", "R"}, Joined → True]]]
]; (*END:SIRModel*)

CompareSimulationWithSIRModel["DefaultArguments"] =
    {"n0" → 1000, "nI0" → 10, "R" → 2., "tMax" → 20, "dtOut" → 1., "nRuns" → 8};

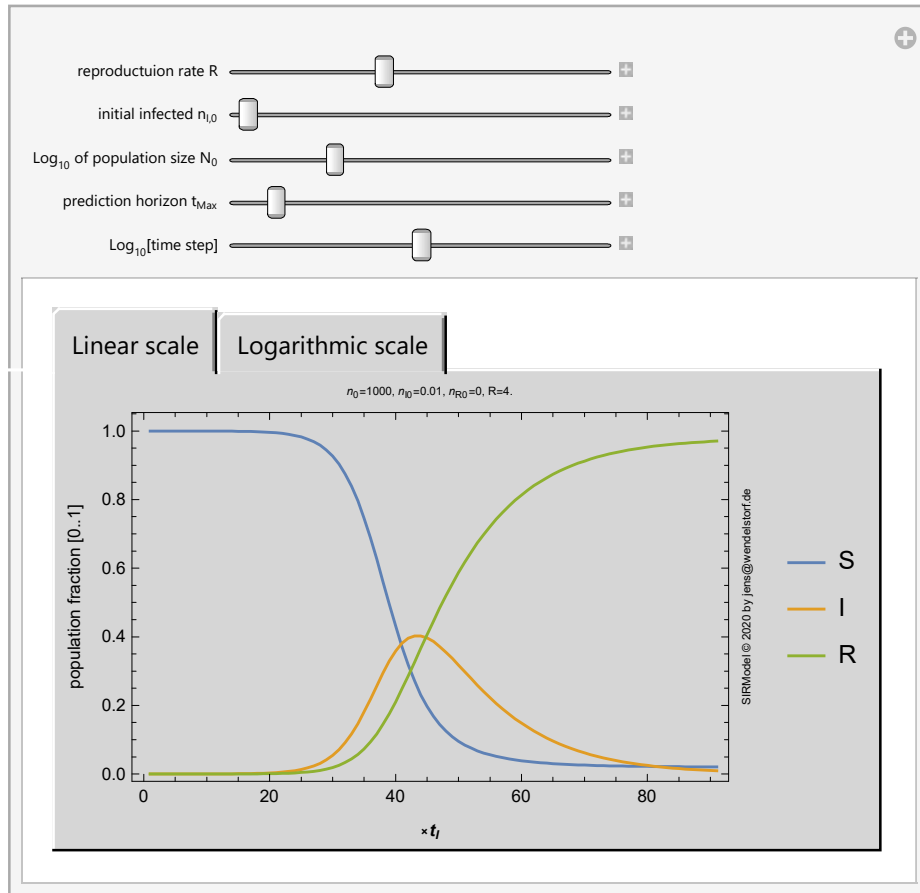
CompareSimulationWithSIRModel[input: {___Rule}] := Module[
{
    inp, pTitle, nRuns,
    rToken, rStatistics, rSimMean, rSimSD, rSim, rSIR,
    x
},
    x = <|Join[CompareSimulationWithSIRModel["DefaultArguments"], input]|>;
    inp = Normal[x]; rToken = "I";
    nRuns = x["nRuns"];
    pTitle = "n0=" <> ToString[x["n0"]] <> ", nI0=" <> ToString[x["nI0"]] <>
        ", R=" <> ToString[x["R"]] <> ", tMax=" <> ToString[x["tMax"]];
    rStatistics = SimulateEpidemicStatistics[nRuns, inp];
    rSimMean = Normal[rStatistics[rToken, "Mean"]];
    rSimSD = Normal[rStatistics[rToken, "SD"]];
    rSim = Table[Around[rSimMean[[i]], 2rSimSD[[i]]], {i, Length[rSimMean]}];
    rSIR = <|SIRModel[inp]|>;
    ListLogPlot[{rSim, rSIR[rToken]},
        Frame → True, FrameLabel → {
            {"I-fraction", Style["CompareSimulationWithSIRModel", Tiny]},
            {"t", pTitle}},
        PlotLegends → {ToString[nRuns] <> " Simulations", "SIR-Model"},
        PlotRange → {0.5*x["nI0"]/x["n0"], 1.}]
]

```

3.1.2 Das interaktive SIR Modell

In[79]:=

Manipulate[...]



Out[79]:=

Dieses Modell liefert glatte Kurven, obwohl ein granulares Problem beschrieben wird (mit $\text{Log}_{10}[\text{time step}]=0$ erhalten Sie nur die Werte zu den auch in der Simulation gelieferten Zeitpunkten). Ein weiteres interaktives SIR Modell finden Sie im Wolfram Demonstrations Project.

3.2 Ein realistischeres SIR - Modell

Wenn Sie ein einfaches Modell haben, welches noch nicht alle gewünschten Aspekte eines Prozesses wiedergibt, so lassen sich die fehlenden Features oft nachträglich hinzufügen. So entstehen komplexe Modelle, die ohne diese schrittweise Vorgehensweise kaum realisierbar sind.

3.2.1 Motivation und Modellparameter

Der Reproduktionsfaktor R ist im Verlauf einer realen Epidemie selten konstant. Führt die Verbreitung eines Krankheitserregers durch die Erkrankten zu gesellschaftlich inakzeptablen Zuständen (Todesfälle), dann kann die Gesellschaft (Population) reagieren. So kommt es in der Population zu Verhaltensänderungen, die zunächst zu einer Abnahme von R führen. Die anfängliche Reproduktionszahl wird Basisreproduktionszahl R_0 genannt. Eine zeitabhängige Reproduktionszahl $R(t)$ ist ein beliebtes Konzept um den Verlauf einer Epidemie zu beschreiben. Dieses Konzept ist jedoch wenig hilfreich, wenn es um Vorhersagen der grundsätzlich auftretenden Phänomene geht. Die Verbreitung über Kontakte und die Populationsgröße granularisieren das Problem, was sich jedoch durchaus in den Gleichungen eines kontinuierlichen Modells berücksichtigen lässt. Gibt es in der Population niemanden mehr, der den Erreger verbreiten kann, so endet die Epidemie auch wenn es noch infizierbare Personen gibt. Ansonsten wird sie wieder aufflammen, wenn der Reproduktionsfaktor durch die Rückkehr zu alten Verhaltensweisen (Kontakte) wieder über $R=1$ ansteigt. Wir suchen hier nach der einfachsten möglichen Modellbeschreibung, die alle wesentlichen Phänomene wiedergibt. Dazu reicht offensichtlich ein konstantes R_0 nicht aus, ein beliebiges $R(t)$ führt jedoch zu einer zu komplexen Aufgabenstellung (die Anzahl der Modellparameter steigt stark an, wenn die Interventionen einzeln parametrisiert werden müssen). Die maximale Vereinfachung ist die Annahme von zwei diskreten R -Werten: Der erste (R) entspricht dem R_0 und der andere (RI) entspricht dem R -Wert, der durch die Interventionen erreichbar ist. Daraus ergeben sich einige zusätzliche Parameter für das Modell:

- **$xdlmax$** beschreibt den Anteil an aktuell Infizierten (0..1), der zu Maßnahmen führt die den Verlauf der Epidemie verlangsamen, d.h. die Reproduktionszahl auf den Wert
- **RI** senken. Ist RI kleiner als 1, so sinkt die Anzahl der Infizierten. Erreicht der Anteil an aktuell Infizierten die Schwelle
- **$xdlmin$** , so werden die Maßnahmen aufgehoben und die Reproduktionszahl erreicht wieder ihren alten Wert R . Falls $xdlmin$ einen Wert überschreitet, der noch mindestens einem infektiösen Fall entspricht, und noch keine Herdenimmunität erreicht wurde, so "geht alles von vorne los". Um diesen Kreislauf zu durchbrechen wird ein Lernfaktor
- **$fLearn$** eingeführt um die Grenzwerte für die Intervention zeitlich variabel zu gestalten. Ein Wert über 1 führt zu größeren (Hybris) und ein Wert unter 1 (Vorsicht) zu kleineren Grenzwerten: Die Parameter $xdlmin$ und $xdlmax$ werden in jedem Zyklus mit **$fLearn$** multipliziert.
- **$IMax$** wird eingeführt um den Zeitraum auszurechnen, in dem das Gesundheitssystem überlastet ist ($I(t) > I_{Max}$).

- **tMax** wird eingeführt um den Zeitraum zu begrenzen, über den der Verlauf einer Epidemie maximal betrachtet wird. Ist nach einem gewissen Zeitraum eine ausreichende Impfquote zu erwarten, so ist dieser Zeitraum ein guter Schätzwert für *tMax*. *tMax* ist auch erforderlich, da für $R \sim 1$ die Epidemie sehr lange andauert.

Das allgemeine Prinzip besteht darin, ein Modell so einfach wie möglich zu gestalten ohne die wesentlichen Phänomene dabei quasi abzuschalten. Einfach bedeutet in der Regel so wenige Parameter wie möglich. Ein zeitabhängiges $R(t)$ kommt somit nicht in Frage, da es quasi beliebig vielen Parametern entspricht. Mit dem obigen Ansatz wird zugleich eine Rückkopplung (feedback) eingeführt, die den Verlauf einer realen Epidemie durchaus wesentlich verändern kann.

Auch wenn das reale, im Nachhinein (a posteriori) berechenbare, $R(t)$ deutlich von dem im Modell verwendeten (abgestuften) $R(t)$ abweicht, so ergibt sich doch eine Parabel auf ein epidemiologisches Modell. Dieses vereinfachte Modell kann dann keine quantitativen Verläufe mehr vorhersagen, gibt jedoch die qualitativen Phänomene und deren Parameterabhängigkeiten gut wieder. Während realistische Modelle schwer nachzuvollziehen und durch eine Vielzahl von Parametern auch nur schwer einzuschätzen sind, können einfache Modelle oft auch einfache Antworten liefern. Wir hatten ja bereits am Anfang die Frage aufgeworfen, ob es sich bei der Beherrschung einer Epidemie um ein Entscheidungsproblem oder ein Regelungsproblem handelt!

Die Umsetzung in ein mathematisches Modell ist nun etwas schwerer und der Quelltext ist sicher nur für Kenner der Wolfram Language lesbar. Das soll jedoch niemanden daran hindern, die Ergebnisse mit Hilfe von interaktiven Visualisierungswerkzeugen selbst nachzuvollziehen.

3.2.1 Der **SIRModelWithNPI**-Quellcode (Wolfram Language)

In[106]:=

```
(* (c) 2020 jens@wendelstorf.de *)
SIRModelWithNPI::usage =
  "SIRModelWithNPI[{parameter→value,...}]
  is an enhanced SIR Model of an epidemic (t=1..tMax)
  using the primary parameters n0, nI0, nR0, R, RI,
  xdImax, xdImin and fLearn.
  It returns the population fractions of S=Susceptible,
  I=Infected, and R=Removed as {[0..1]} vectors of length tMax.
  tMax is the duration of the epidemic which is also returned.";

SIRModelWithNPI["DefaultArguments"]={
  "n0" → 10000, "nI0" → 1, "nR0" → 0,
  "IMax" → 0.01, (* health system capacity *)
  "dtOut" → 0.1, "fLearn" → 1., "R" → 4., "RI" → 0.5,
  "xdImax" → 5./100., "xdImin" → 1./100000., "tMax" → 1000.
};

SIRModelWithNPI[input:{__Rule}] := Module[
{
  alpha, dtOut, detectStop, eqns,
  fS, fI, fR, fS0, fI0, fR0,
  fIZero, fLearn, fyalpha, mode, n0,
  pRact, pR, pRI, pRList, pRold, s,
  t, tMaxIntegration, tStop, tChange, tNPI,
  tIMax, vars,
  x, xNPION, xNPIOff, xNPI,
  yS, yI, yR, yalpha
},
x = <|Join[SIRModelWithNPI["DefaultArguments"],input]|>;
dtOut = x["dtOut"]; fLearn = x["fLearn"];
tMaxIntegration = x["tMax"]; tStop=tMaxIntegration;
n0 = x["n0"]; fI0 = N[x["nI0"]/n0];
fR0 = N[x["nR0"]/n0]; fS0 = 1. - fI0 - fR0;
pR = x["R"]; pRI = Min[x["RI"],pR];
fIZero = 0.5/n0; pRact = pR; tChange=1.;
xNPION = Max[x["xdImax"],2*fI0]; xNPIOff = Min[x["xdImin"], xNPION];
mode = 0; detectStop=False; pRList = {{1., pRact},{1., pRact}};
xNPI := If[mode==0, xNPION, xNPIOff];
alpha[t_?NumericQ] := Module[
(* we need some bookkeeping for remembering R(t) *)
{ },
If[pRList[[-1,2]] ≠ pRact,
  pRList[[-1]] = {tChange-dtOut/10, pRold};
  AppendTo[pRList, {tChange, pRact}];
  AppendTo[pRList, {t, pRact}],
(*else*)
  pRList[[-1,1]] = t;
];
pRact];
eqns = { (* differential equations for fS, fI, fR: *)
  fS'[t] == -alpha[t]*fS[t]*fI[t],
  fI'[t] == alpha[t]*fS[t]*fI[t] - fI[t],
  fR'[t] == fI[t], (* check R+I+S=n0 *)
  (* initial conditions for fS, fI, fR: *)
  fS[1.] = fS0, fI[1.] = fI0, fR[1.] = fR0,
  (* events controlling the reproduction rate ToDo: check tMaxResult*)
  (*
  WhenEvent[fI[t] - fIZero == 0, If[detectStop,tStop=t; "StopIntegration"],
    "LocationMethod"→"Brent"],
  *)
  WhenEvent[fI[t] < fIZero, {tStop=t, "StopIntegration"}],
```

```

    "LocationMethod"→"Brent"),
  WhenEvent[fI[t] - xNPI == 0, {
    detectStop = True; tChange=t; pRold=pRact;
    If[mode==0, pRact=pRI;mode=1, pRact=pR;mode=0],
    {xNPIOn, xNPIOff} *= fLearn,
    "CrossDiscontinuity"}]
  };
vars = {fS, fI, fR};
s = First[NDSolve[eqns, vars, {t, 1., tMaxIntegration},
  StartingStepSize → dtOut, MaxStepSize → dtOut]];
If[pRList[[-1]]==pRList[[-2]],pRList[[-1]]= Nothing];(* remove duplicate last element *)
tNPI = Total[Table[If[pRList[[i,2]]<pR && pRList[[i+1,2]]<pR,
  pRList[[i+1,1]]-pRList[[i,1]], Nothing],
  {i, 1, Length[pRList]-1}]];
tIMax = Total[Table[
  If[Evaluate[fI[t] /. s] ≥ x["IMax"], dtOut, Nothing],
  {t, 1., tStop, dtOut}]];
(* scale R(t) to 1=R(0): *)
pRList = {First[#],Last[#]/pR} & /@ pRList;
fyalpha = Interpolation[pRList, InterpolationOrder → 1];
{yS, yI, yR, yalpha} = Table[{t,
  Evaluate[# [t] /. s]}, {t, 1., tStop, dtOut}]
  & /@ {fS, fI, fR, fyalpha};
{
  "S" → yS, "I" → yI, "R" → yR, "alpha" → yalpha,
  "tMax" → tStop, "tNPI" → tNPI, "tIMax" → tIMax,
  "IMax" → Max[yI[[All,2]]], "RMax" → Last[Last[yR]]
}
];(* END:SIRModelWithNPI *)

SIRModelWithNPVisualization[input:{___Rule}]:=Module[
  {fLabel, pTitle, pRangeLog, r, x},
  x = <|Join[SIRModelWithNPI["DefaultArguments"], input]|>;
  r = <|SIRModelWithNPI[input]|>;
  pRangeLog = {N[1/(2 x["n0"])],1};
  pTitle = "n0="<>ToString[N[x["n0"]]]<>
    ", nI0="<>ToString[N[x["nI0"]]]<>, nR0="<>ToString[x["nR0"]]<>
    ", R0="<>ToString[x["R"]]<>, RI="<>ToString[x["RI"]]<>
    ", xdi_min="<>ToString[FortranForm[x["xdImin"]]]<>
    ", xdi_max="<>ToString[FortranForm[x["xdImax"]]]<>
    ", fLearn="<>ToString[x["fLearn"]]<>, tMax="<>ToString[N[r["tMax"]]];
  fLabel = {"population fraction [0..1]",
    Style["SIRModelWithNPI © 2020 by jens@wendelstorf.de",Tiny]},
    {"*tI", Style[pTitle,Tiny]};
  Off[Interpolation::inhr];
  TabView[{
    "Linear scale" → ListPlot[{r["S"], r["I"], r["R"], r["alpha"]},
      Frame → True, PlotRange → All, FrameLabel → fLabel,
      PlotLegends → {"S", "I", "R", "α/α₀"}, Joined → True],
    "Logarithmic scale" → ListLogPlot[{r["S"], r["I"], r["R"], r["alpha"]},
      Frame → True, PlotRange → pRangeLog, FrameLabel → fLabel,
      PlotLegends → {"S", "I", "R", "α/α₀"}, Joined → True],
    "Data"→Dataset[EpidemicOutcome[input]]
  ]}
];(*END:SIRModelWithNPVisualization*)

ClearAll[EpidemicOutcome];(* functions with memory *)

EpidemicOutcome["DefaultArguments"] = Join[
  SIRModelWithNPI["DefaultArguments"],
  {
    "fRIOutcome" → 1., "fRMaxOutcome" → 1.,
    "fIMaxOutcome" → 0., "ftMaxOutcome" → 0.
  }
];

```



```

EpidemicOutcome[input:{__Rule}] := Module[
  {r, x, yNPIOOutcome, yRMaxOutcome, yIMaxOutcome, ytMaxOutcome},
  r = <|SIRModelWithNPI[input]|>;
  x = <|Join[EpidemicOutcome["DefaultArguments"],input]|>;
  yNPIOOutcome = x["fRIOOutcome"]*r["tNPI"]*(-Log[x["RI"]/x["R"]]);
  yIMaxOutcome = x["fIMaxOutcome"]*r["tIMax"]*(1+Tanh[4(r["IMax"]-x["IMax"])/x["IMax"]])/2;
  yRMaxOutcome = x["fRMaxOutcome"]*r["RMax"];
  ytMaxOutcome = x["ftMaxOutcome"]*r["tMax"];
  <|"NPIOOutcome" → yNPIOOutcome, "RMaxOutcome" → yRMaxOutcome,
    "tMaxOutcome" → ytMaxOutcome, "IMaxOutcome" → yIMaxOutcome,
    "TotalOutcome" → yNPIOOutcome+yRMaxOutcome+ytMaxOutcome+yIMaxOutcome,
    "RMax" → r["RMax"], "tNPI" → r["tNPI"],
    "tMax" → r["tMax"], "tIMax" → r["tIMax"], "IMax" → r["IMax"] |>
];

VisualizeEpidemicOutcome["DefaultArguments"] = {
  "fRIOOutcome" → 1., "fRMaxOutcome" → 1.,
  "fIMaxOutcome" → 0., "ftMaxOutcome" → 0.,
  "LogScale" → False, "p" → "RI", "pMin" → 0.1, "pMax" → 1.5
};

VisualizeEpidemicOutcome[input:{__Rule}] := Module[
  {doPlot, f, pIterator, x, xArgs},
  xArgs = <|Join[VisualizeEpidemicOutcome["DefaultArguments"],input]|>;
  f[inp_?NumericQ] := f[inp] = EpidemicOutcome[Join[input,{xArgs["p"]→inp}]];
  Which[xArgs["LogScale"]==True,
    pIterator = {x, 10^xArgs["pMin"], 10^xArgs["pMax"]};
    doPlot[v_] := v → LogLinearPlot[ f[x][v], Evaluate[pIterator],
      GridLines → Automatic, Frame → True,
      PlotRange → Full, FrameLabel → {xArgs["p"],v}];
  ,True,
    pIterator = {x, xArgs["pMin"], xArgs["pMax"]};
    doPlot[v_] := v → Plot[ f[x][v], Evaluate[pIterator],
      GridLines → Automatic, Frame → True,
      PlotRange → Full, FrameLabel → {xArgs["p"],v}];
  ];
  TabView[doPlot /@ {
    "TotalOutcome", "RMax",
    "tMax", "tNPI", "tIMax", "IMax",
    "RMaxOutcome", "IMaxOutcome", "NPIOOutcome", "tMaxOutcome"}
  ]
];

MinimizeEpidemicOutcome["DefaultArguments"] = {
  "fRIOOutcome" → 1., "fRMaxOutcome" → 200.,
  "fIMaxOutcome" → 0., "ftMaxOutcome" → 0.
};

MinimizeEpidemicOutcome[input:{__Rule}] := Module[
  { f, p, n, xArgs, y, yMin, yMax },
  xArgs = <|Join[SIRModelWithNPI["DefaultArguments"],
    MinimizeEpidemicOutcome["DefaultArguments"],input]|>;
  yMin = xArgs["RI"];
  yMax = xArgs["R"];
  f[p_?NumericQ] := f[p] = EpidemicOutcome[Join[input,{"RI"→p}]]["TotalOutcome"];
  "RI" → First[Values[Last[NMinimize[{f[y],yMin≤y≤yMax}, y]]]]
];

(* EOF: WL Quellcode (c) 2020 by jens@wendelstorf.de *)

```

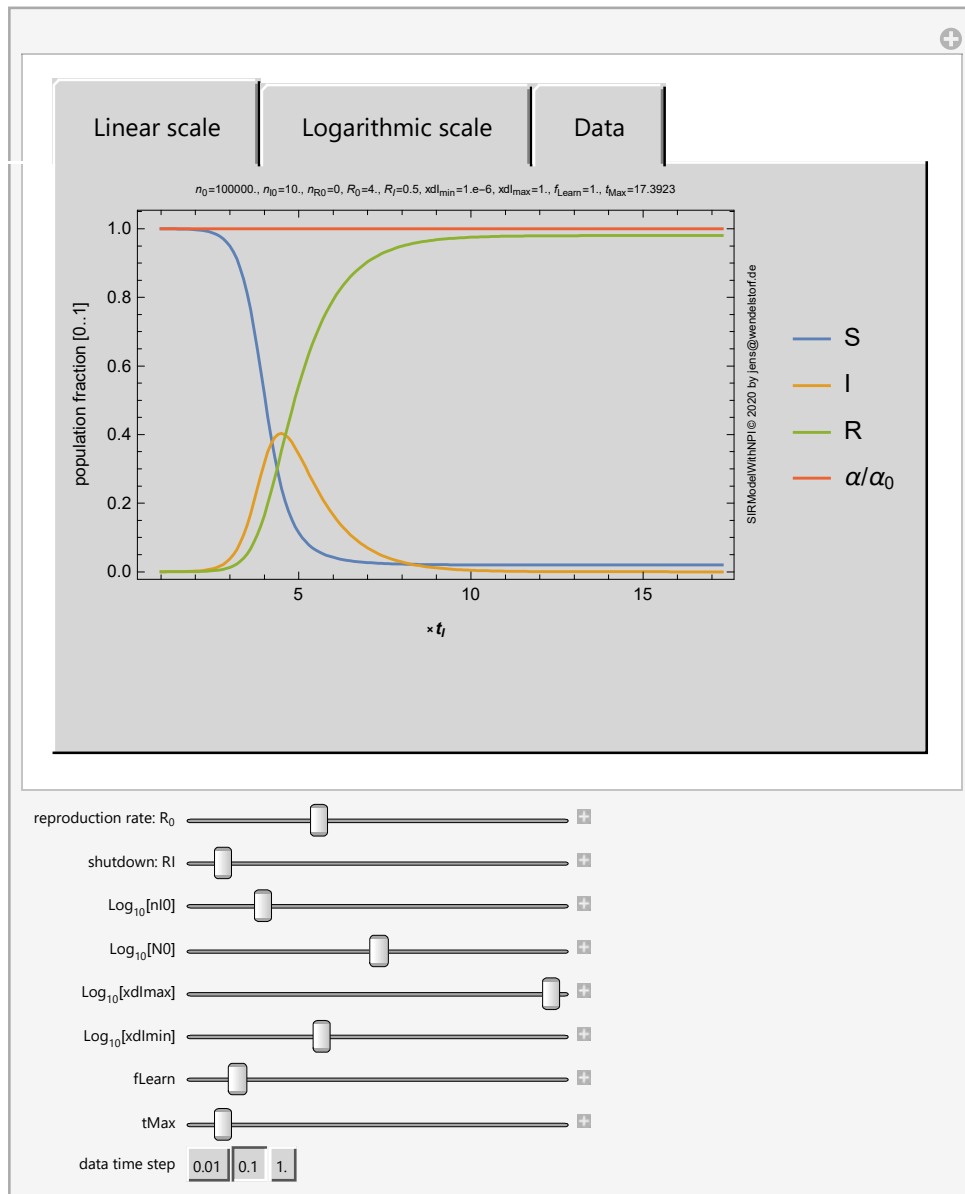
3.3 Die interaktive Berechnung des **Verlaufes** einer Epidemie

Mit unserem erweiterten SIR Modell können Sie nun dieselben Parameterstudien durchführen, wie mit dem Simulationsmodell aus Abschnitt 2, lediglich die Rolle des Zufalls bleibt unberücksichtigt.

In[80]:=

Manipulate[...]

Out[80]=



Achtung:

Nicht alle Parametereinstellungen sind sinnvoll!

Versuchen Sie zu **verstehen** was Sie tun, es werden im Folgenden nicht alle Einstellungen angegeben.

Durch Klicken auf das Plus-Zeichen erhalten Sie jeweils genauere Einstellmöglichkeiten!

Was können Sie wie erreichen:**1. $\text{Log}_{10}[\mathbf{xdlmax}]$ verkleinern:**

Wenn der Schwellwert **xdlMax** für eine Intervention erreicht wird, tritt **RI** in Aktion!

1.1. Bei $\text{Log}_{10}[\mathbf{xdlMax}]=-1$ erfolgt die Intervention wenn 10% der Population infiziert sind: Über 20% Betroffene (grüne R-Kurve) und die Epidemie ist nach $17.4 \cdot t_i$ Infektionsperioden vorbei (für $N_0=100.000$).

1.2. Bei $\text{Log}_{10}[\mathbf{xdlMax}]=-3$ schon wenn 0.1% der Population infiziert sind:
Weniger als 0.3% Betroffene (grüne R-Kurve) und die Epidemie ist nach etwas über $12 \cdot t_i$ Infektionsperioden vorbei.

2. $\text{Log}_{10}[\mathbf{xdlmin}]$ vergrößern:

Wenn der Schwellwert für die Beendigung des Lockdowns mindestens einen Infizierten übrig lässt, verläuft die Epidemie in Wellen:

2.1. Bei $\text{Log}_{10}[\mathbf{n0}]=5$, d.h. einer Populationsgröße von 100.000 treten die Wellen ab $\text{Log}_{10}[\mathbf{xdlMin}]>-6$ auf und schon $\text{Log}_{10}[\mathbf{xdlMin}]=-5$ und $\text{Log}_{10}[\mathbf{xdlMax}]=-1$ führt zu drei Infektionswellen mit insgesamt 68% Betroffenen.

2.2. Wenn Sie die Populationsgröße verringern ($\text{Log}_{10}[\mathbf{N0}]=3$ bedeutet **N0**=1000) wird **xdlmin** wieder klein genug und die Epidemie ist schneller vorbei.

3. **fLearn verändern:**

Hybris (**fLearn**>1) und **xdlmin** und **xdlmax** werden nach jeder Welle vergrößert bzw. Vorsicht (**fLearn**<1) und **xdlmin** und **xdlmax** werden nach jeder Welle verkleinert:

3.1. **fLearn**=2 und die zweite Welle wird größer als die erste.

3.2. **fLearn**=0.5 und jede Welle wird kleiner als die vorangegangene.

4. **tMax verändern:**

Die Dauer der Epidemie wird nur richtig berechnet, wenn diese kleiner als das vorgegebene **tMax** ist!

3.4 Die Ergebnisse einer Epidemie

Die Simulation des zeitlichen Verlaufes einer Epidemie liefert eine Vielzahl von Daten. Von einigen Kurven ist nur der Maximalwert interessant und es lässt sich eine Reihe von Kennzahlen für die Epidemie als Ganzes extrahieren (**EpidemicOutcome[...]**-Funktion):

- **tMax** ist die Dauer der Epidemie in Infektionsperioden (bis keine Neuinfektionen mehr möglich sind).
- **RMax** ist der Anteil der Population, der von der Epidemie betroffen ist.
Für große **R** und ohne nachhaltige Intervention sind das fast 100%!
- **IMax** ist der Maximalwert für die aktuell Infizierten. Als Parameter gleichen Namens bestimmt **IMax** die Zeiträume, an denen die Kapazität des Gesundheitssystems überschritten wird.
- **tIMax** ist der Zeitraum, in dem **I(t)** über **IMax** liegt.

- **tNPI** ist der Zeitraum, in dem das *Interventions-R RI* aktiv ist
- **NPIOutcome** ist eine Kostenfunktion, die über das Gewicht **fRIOutcome** beeinflusst werden kann. Hier werden die Zeiträume, in denen durch Intervention (Lockdown) *R* auf *RI* gesetzt wird, addiert.
- **RMaxOutcome** ist eine Kostenfunktion, die über das Gewicht **fRMaxOutcome** beeinflusst werden kann. Hier wird der von der Epidemie Betroffene Populationsanteil berücksichtigt
- **tMaxOutcome** ist eine Kostenfunktion, die über das Gewicht **ftMaxOutcome** beeinflusst werden kann. Hier wird die Gesamtdauer der Epidemie berücksichtigt.
- **IMaxOutcome** ist eine Kostenfunktion, die über das Gewicht **fIMaxOutcome** beeinflusst werden kann. Hier werden die Zeiträume, in denen die Kapazität des Gesundheitssystems (Parameter **IMax**) nicht ausreicht, berücksichtigt.
- **TotalOutcome** ist die Summe aller Kostenfunktionen.

Die quantitativen Ergebnisse können Sie sich oben im Werkzeug von Abschnitt 3.3 mit dem Reiter Data anzeigen lassen.

3.5 Die Analyse von Parameterabhängigkeiten mit Hilfe einer Kostenfunktion

Hat man erst einmal gelernt, welche Ergebnisse eines Modells relevant sind, so lassen sich aus diesen Ergebnissen Größen ableiten, die ein spezifisches Ziel quantifizieren. Als Beispiel dienen hier Quantifizierungen, die mit Hilfe der Funktion **EpidemicOutcome[..]** bereitgestellt werden:

- Das **NPIOutcome** ist das Produkt aus dem Faktor **fRIOutcome** und der Zeitdauer der Interventionen, wobei nach der Intensität der Intervention (*RI*) gewichtet wird:

$$NPIOutcome = fRIOutcome * tNPI * (-Log[RI/R])$$
- Das **IMaxOutcome** quantifiziert den Zeitraum, in dem das Gesundheitssystem überlastet wird:

$$IMaxOutcome = fIMaxOutcome * tIMax * (1 + Tanh[4 * (IMax_{Ist} - IMax_{Grenzwert}) / IMax_{Grenzwert}]) / 2$$
- Das **RMaxOutcome** quantifiziert den Anteil der Population, der letztendlich direkt von der Epidemie betroffen wird (\Rightarrow die über die CFR bestimmte Anzahl der Toten):

$$RMaxOutcome = fRMaxOutcome * RMax$$
- Das **tMaxOutcome** quantifiziert die Dauer der Epidemie und der damit verbundenen qualitativen Unterschiede zum Normalzustand:

$$tMaxOutcome = ftMaxOutcome * tMax$$

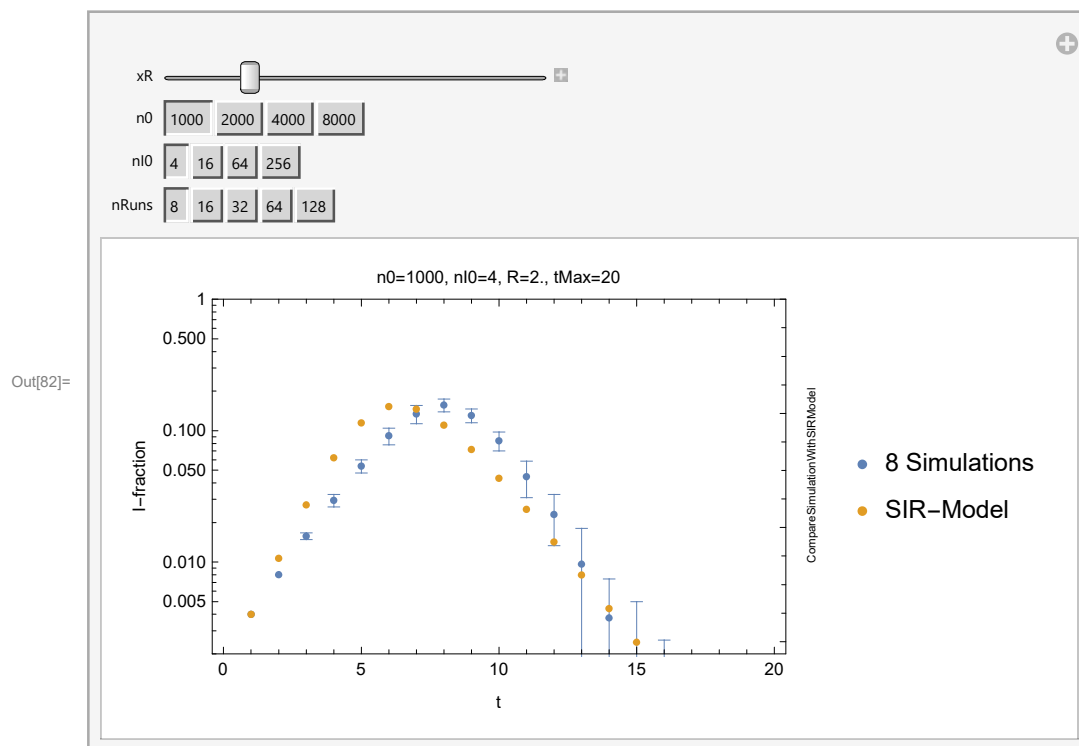
Da eine Mehrzieloptimierung in der Regel schwierig ist, lassen sich die obigen Quantifizierungen zu einer gewichteten Summe addieren, die von **EpidemicOutcome[..]** als **TotalOutcome** geliefert wird. In Abschnitt 6 werden wir dieses Werkzeug verwenden um die Beeinflussung einer Epidemie zu diskutieren.

4. Vergleich der Modelle

In den beiden vorangegangenen Abschnitten habe wir zwei grundlegend unterschiedliche Modelle für eine Epidemie kennengelernt, die wir nun miteinander vergleichen wollen. Das Simulationsmodell stellt den Versuch dar, die Neuinfektionen durch zufällige Kontakte zu simulieren. Das klassische SIR Modell mit Differentialgleichungen beschreibt einen kontinuierlichen Verlauf einer Epidemie und liefert, wie im folgenden demonstriert wird, qualitativ ähnliche Ergebnisse wie das Simulationsmodell. Es kann jedoch die statistischen Schwankungen nicht abbilden und sollte daher auch nur mit den Mittelwerten aus einer großen Anzahl von Simulationen verglichen werden (die Simulationen können durchaus etwas Rechenzeit beanspruchen!):

```
In[81]:= ClearAll[fCmpPlot];
fCmpPlot[x_] := fCmpPlot[x] = CompareSimulationWithSIRModel[x];
```

Manipulate[...]



■ **Bemerkung:**

Die qualitative Äquivalenz ist sicher eine glaubhafte Hypothese, die quantitative erfordert etwas Feinarbeit an den Modellen!?

5. Diskussion der Parameter

Modelle sollen alles berücksichtigen, was die vorherzusagenden Größen signifikant beeinflusst. Bei komplexen Prozessen führt das zu komplexen Modellen, die historisch gewachsen und nur noch schwer zu überblicken sind. Leider ist es nicht immer möglich, quantitativ richtige Vorhersagen aus einfachen Modellen zu gewinnen. Hier helfen Verständnismodelle, die vor allem das qualitative Verhalten eines Prozesses abbilden. Wir wollen uns nun ansehen, wie die wesentlichen Phänomene eines Epidemieverlaufes in unseren Verständnismodellen abgebildet werden.

Die Reproduktionszahl R

Eine Epidemie kann sich sehr schnell in einer Population verbreiten oder aber auch gar nicht. Diese Intensität des Ansteckungsgeschehens hängt von den Mechanismen der Verbreitung des Erregers und den Maßnahmen ab, mit denen versucht wird das Geschehen zu beeinflussen. Zur Quantifizierung dient uns die Reproduktionszahl R . Diese findet sich sowohl in dem stochastischen Simulationsmodell als auch in dem mathematisch formulierten SIR-Modell. In komplexen Populationen wird dieses R eine zeitabhängige Größe $R(t)$, die im Nachhinein aus den Verlaufsgrößen bestimmt wird. Hier tritt in der Realität eine Rückkopplung auf, deren Vorhersage nur möglich wäre, wenn die Einflussfaktoren und Regelkreise vorab feststehen würden. So ist die Nichtvorhersagbarkeit von Epidemien in komplexen menschlichen Gesellschaften [7] wohl ein durch deren Organisation bedingtes Phänomen, welches in sehr einfachen Populationen nicht auftritt (natürlicher, d.h. u.U. fataler Verlauf - siehe Abschnitt 6.1).

In den Modellen verzichten wir insofern auf eine quantitative Verlaufsprognose, als das wir $R(t)$ nur zwischen diskreten Werten hin- und herspringen lassen:

- am Anfang ist $R = R_0 = R(t = 0)$.
- überschreitet der Anteil an aktuell Infizierten in der Population einen Grenzwert x_{dlmax} , so wird durch Interventionsmaßnahmen (Kontaktbeschränkungen, Lockdown) die Reproduktionszahl auf R_l reduziert.
- unterschreitet der Anteil an aktuell Infizierten in der Population einen Grenzwert x_{dlmin} , so werden die Maßnahmen aufgehoben und die Reproduktionszahl erreicht wieder ihren alten Wert R .
- Um dieses Hin- und Herspringen von R noch zu verfeinern wird zusätzlich ein Faktor f_{Learn} eingeführt mit dem die Grenzwerte für die Intervention jeweils multipliziert werden. Ein Wert über 1 führt zu größeren (Hybris) und ein Wert unter 1 (Vorsicht) zu kleineren Grenzwerten.

Der Anfangszustand

Der Verlauf einer Epidemie hängt nicht nur von den durch R beschriebenen Verbreitungsmechanismen sondern auch vom Anfangszustand der Population ab:

- n_0 beschreibt die Größe der Population und bestimmt damit die stochastische Variabilität des Verlaufes und das Ende der Epidemie (Problemgranularität).
- n_{i0} ist die Anzahl der Anfangs Infizierten und
- n_{r0} ist die Anzahl der Anfangs bereits nicht mehr Infizierbaren.

Modellparameter

Nicht alle Parameter eines Modells gehören zwingend auch zum jeweiligen Prozess. Oft soll lediglich die konkrete Berechnung durch Parameter beeinflusst werden:

- ***I*Max** wird eingeführt um den Zeitraum auszurechnen, in dem das Gesundheitssystem überlastet ist ($I(t) > I_{Max}$).
- ***t*Max** wird eingeführt um den Zeitraum zu begrenzen, über den der Verlauf einer Epidemie maximal betrachtet wird. Ist nach einem gewissen Zeitraum eine ausreichende Impfquote zu erwarten, so ist dieser Zeitraum ein guter Schätzwert für *t*Max. *t*Max ist auch erforderlich, da für $R \sim 1$ die Epidemie sehr lange andauert.

Anpassungsparameter

Das Wort Anpassungs- oder Fit-Parameter haben Sie sicher schon gehört. Anpassungsparameter werden verwendet, wenn eine Vorhersage nur auf der Basis von Naturgesetzen unrealistisch ist. Die Werte von Anpassungsparametern werden durch den Vergleich der Berechnungsergebnisse mit Messungen gewonnen. Die Suche nach Modellen, die nicht im Widerspruch zu einer vorhandenen Menge an Messungen stehen, bezeichnet man als **Modellidentifikation**. Es handelt sich um eine schwierige und fehleranfällige Aufgabe. Ist es (zumindest theoretisch) möglich die Anpassungsparameter eines Modells zu bestimmen, so spricht man von Identifizierbarkeit. Das Problem der Vorhersagegenauigkeit kann nicht einfach durch die Einführung immer neuer Anpassungsparameter gelöst werden, da diese dann nicht mehr eindeutig bestimmbar sind. Ein Modell muss identifizierbar (parametrierbar) sein. Dies ist ein wichtiges Thema der wissenschaftlichen Prozessmodellierung.

Da alle Messungen mit Messunsicherheiten verknüpft sind, können auch die Anpassungsparameter nicht exakt bestimmt werden. Das Konzept einer *Messunsicherheit für Anpassungsparameter* ist nur für Modelle mit genau einem Parameter zielführend. Für Modelle mit mehreren Anpassungsparametern ist zumindest die *Region der durch die Messungen nicht falsifizierten Parametersätze* zu bestimmen. Diese Region ist wesentlich kleiner als sie einhüllende Körper im Parameterraum, wie zum Beispiel das einer "Parameterspezifikation mit Fehlerbalken" entsprechende Hyperrechteck oder eine oft als Spezifikation verwendete Hyperellipse [9].

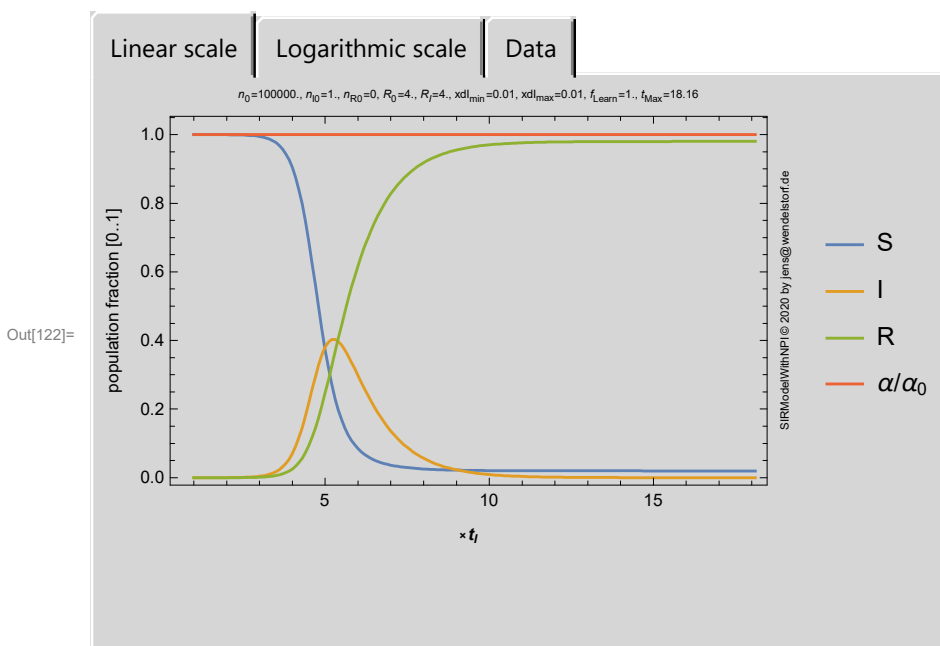
6. Ergebnisse

Wie schon erwähnt bezwecken unsere Modelle hier nicht die konkrete Vorhersage einer Epidemie - ein derartiges Unterfangen ist nur möglich, wenn die Reaktion der Population auf die Epidemie a priori feststeht. Trotzdem geben die Modelle die wesentlichen Phänomene wieder, die für das Verständnis einer Epidemie und damit für deren Eingrenzung bedeutsam sind. Sie sind aufgerufen, die Beispiele als Anregung für eigene Rechnungen zu betrachten.

6.1 Einige Szenarien

■ Der *natürliche* Lauf der Dinge ($R = 4$):

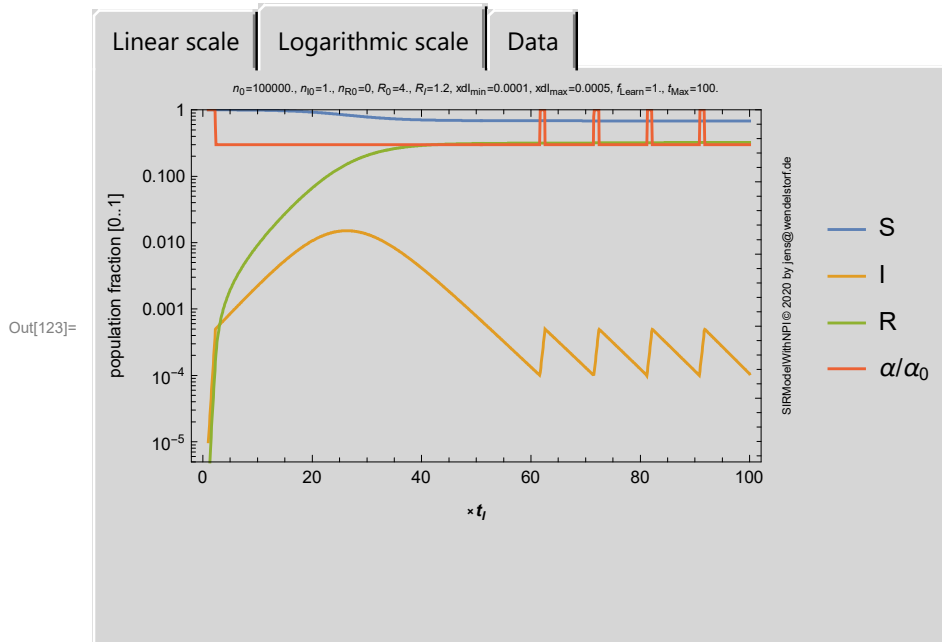
```
In[122]:= SIRModelWithNPVisualization[
  {"n0" → 100000., "RI" → 4., "xdImin" → 0.01, "xdImax" → 0.01}]
```



■ Die ewige Intervention ($R = 4$, $RI=1.2$):

In diesem Beispiel wird schnell reagiert, der Interventionen reduzieren die Reproduktionszahl jedoch nicht unter den Wert von 1 und die Interventionen werden zu früh aufgehoben obwohl es noch Infektionen gibt. Damit gibt es zwar nur eine Welle, die Epidemie wird jedoch nicht beendet sondern es wird lediglich der Zeitraum bis zur Durchseuchung der Population extrem verlängert.

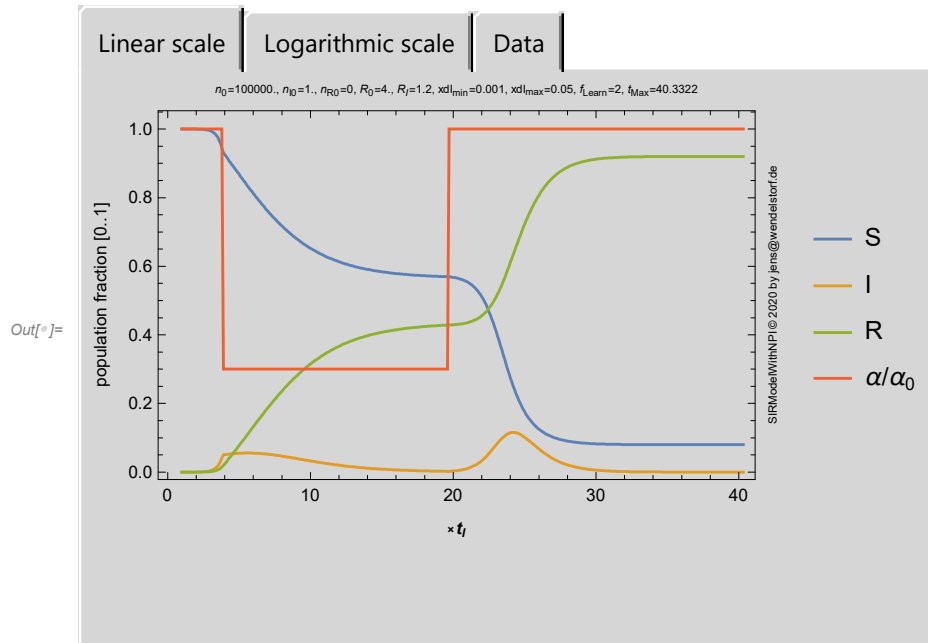
```
ln[123]:= SIRModelWithNPVisualization[{"n0" -> 100 000., "R" -> 4.,
  "RI" -> 1.2, "tMax" -> 100, "xdImin" -> 10 / 100 000, "xdImax" -> 50 / 100 000} ]
```



■ Hybris bringt die 2. Welle ($R = 4$, $RI = 1.2$):

Die erste Intervention wird im nachhinein als übertrieben eingeschätzt, obwohl sie zu schwach war. Als Ergebnis fällt die 2. Welle intensiver als die 1. aus und letztendlich ist wieder die ganze Population betroffen.

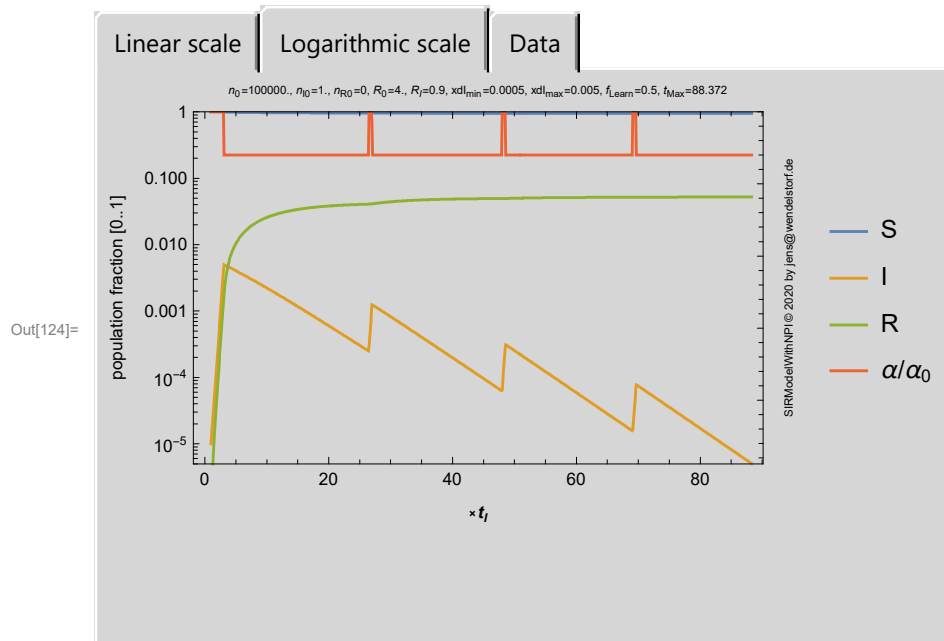
```
In[*]:= SIRModelWithNPVisualization[
  {"n0" -> 100000., "RI" -> 1.2, "fLearn" -> 2, "xdImin" -> 0.001, "xdImax" -> 0.05}]
```



■ Lernfähiger Manager:

Wenn der Interventionspunkt ($x_{d\max}$) konservativ gesetzt ist und konsequent alle Parameter ($x_{d\min}, x_{d\max}$) schrittweise verschärft werden, so wird letztendlich auch ein Ende der Epidemie erreicht. In diesem Fall sorgt $f_{\text{Learn}}=0.5$ für eine Halbierung von $x_{d\max}$ und damit ein Durchhalten der Interventionen bis zum letzten Infizierten.

```
In[124]:= SIRModelWithNPVisualization[{"n0" -> 100 000., "RI" -> 0.9, "fLearn" -> 0.5,
  "xdImin" -> 50 / 100 000, "xdImax" -> 500 / 100 000}]
```

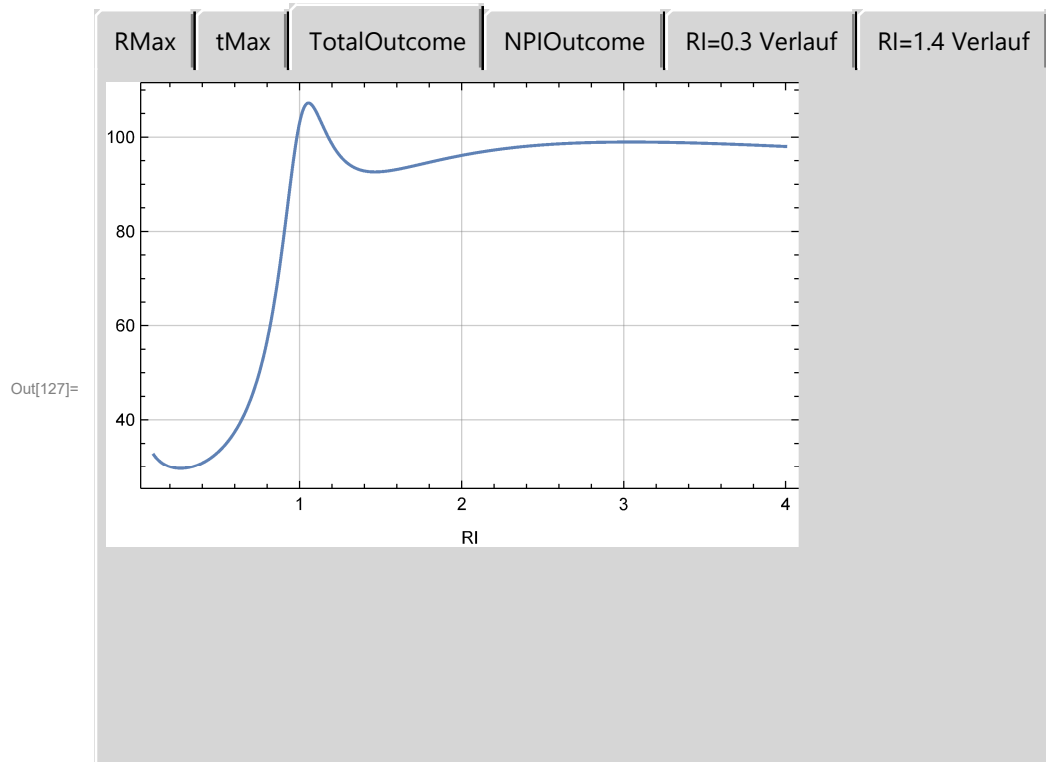


- zugleich ist zu erkennen, wie essentiell die Aufteilung von großen Populationen ist um wirklich das Epidemieende zu erreichen!?

■ Welche Intervention liefert minimale Kosten:

Mit einer Kostenfunktion kann versucht werden eine optimale Interventionsintensität **RI** zu finden.

```
In[125]:= n0 = 100000; inp = { ... }; ClearAll[f];
f[ri_?NumericQ] := f[ri] = EpidemicOutcome[Join[{"RI" -> ri}, inp]];
SetOptions[ ... ];
TabView[ { ... } ]
```



6.2 Eigene Experimente

Indem Sie mit dem Modell experimentieren, können Sie etwas über die Phänomenologie einer Epidemie lernen:

- Eine Epidemie hat einen **natürlichen Verlauf** (konstantes R) bei dem ein maximaler Anteil der Population betroffen ist, z.B. 80% bei $R=2$ und mehr als 95% für $R>3$.
- Eine Epidemie endet, wenn dieser maximale Anteil erreicht ist oder wenn es keine Neuinfektionen mehr gibt. Letzteres Ereignis hängt von der Populationsgröße ab und unterliegt statistischen Schwankungen.
- Die (anfängliche) Reproduktionsrate R und R_I bestimmen die Zeitskala, wobei natürlich immer in Vielfachen von t_I gerechnet werden muss.
- Lockerungen (von Maßnahmen zur Reduktion von R) können auch zu früh kommen, dann verläuft die Epidemie in Wellen:
Vergleichen Sie $x_{dlmin}=10^{-5}$ mit $x_{dlmin}=10^{-4}$ und setzen Sie **fLearn** auf 1 (kein Lerneffekt)! Erst wenn die Population frei von Infektiösen ist, ist die Epidemie zu Ende. Zwischen den Wellen vergeht, in Abhängigkeit von R_I , viel Zeit.
- Lockerungen können auch übertrieben sein (**fLearn**>1), dann ist die zweite Welle heftiger als die erste.
- Ein Reproduktionsfaktor von $R=1$ führt zu einer sehr langen Epidemie.
- Das entscheidende Ergebnis einer Epidemie ist der Anteil der am Ende Betroffenen. Um diesen zu minimieren, muss die Reproduktionsrate bis zum völligen Verschwinden der Neuinfektionsrate auf einem Wert deutlich unter 1 gehalten werden.

6.3 Wie lässt sich eine Epidemie optimal beeinflussen?

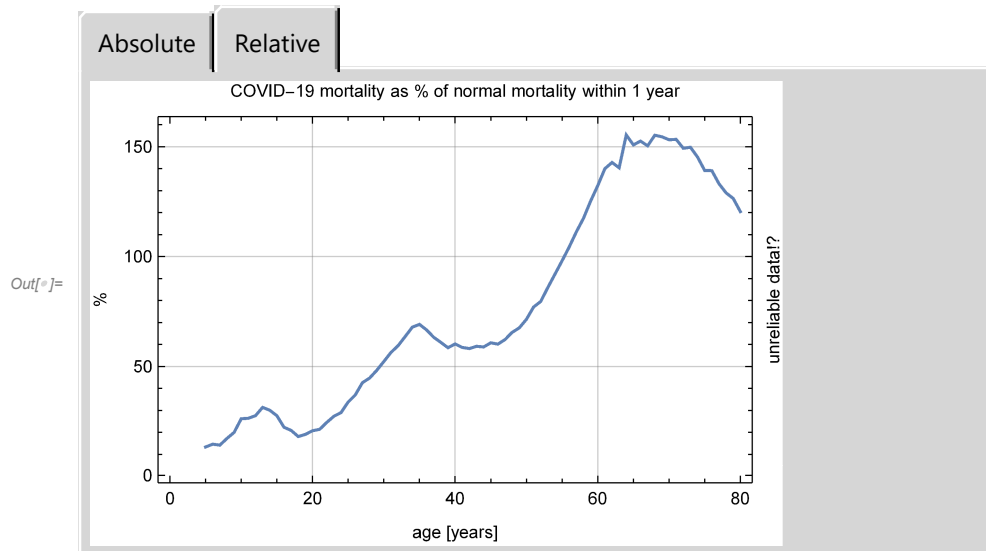
Der Verlauf einer Epidemie hängt von zwei grundsätzlich zu unterscheidenden Kategorien von Faktoren ab:

6.3.1 Krankheits- und Überträgerfaktoren

Ohne Ansteckungsereignisse gibt es keine Epidemie. Die bereits diskutierte Reproduktionszahl R wird nicht nur von der Anzahl und Art der Kontakttereignisse sondern auch von der Übertragungsart des Krankheitserregers bestimmt. Jede Infektion führt zu einem gewissen Zeitraum in dem der Infizierte die Krankheitserreger weitergeben kann. Dabei ist schon der Begriff Krankheitserreger irreführend, da viele Mikroorganismen nur bei einem Teil der Infizierten überhaupt eine merkliche Erkrankung auslösen. Sind diese Mikroorganismen für einen Teil der Population jedoch gefährlich, so kann sich eine tückische, anfänglich schwer zu bemerkende Epidemie ergeben. Letztendlich muss somit am Anfang einer Epidemie vor allem der Erreger gefunden werden. Eine Epidemie lässt sich somit auf Grund von Eigenschaften des Erregers oder der Verteilung der möglichen Krankheitsverläufe immer nur in begrenztem Umfang beeinflussen. So sind beim **SARS-CoV-2** Virus am Anfang nur die Experten (Virologen) entscheidend, da ja zunächst entschieden werden muss, ob sich ein Virus leicht verbreitet und ob es eine Bedrohung für relevante Teile der Population darstellt. Die Größe der Bedrohung wird durch die sogenannte **Case Fatality Ratio (CFR)** und die **Infection Fatality Ratio (IFR)** angegeben, die beispielsweise bei einer Grippeepidemie über alle

Altersgruppen gemittelt 0.1% beträgt. Dabei ist auch die Verteilung über die Altersgruppen wichtig. Bei COVID-19 liegen erste Daten vor [10], die mit der Grundmortalität verglichen werden können:

```
In[ ]:= Covid19IFRdata = { ... } ;
fCovid = Interpolation[Covid19IFRdata]; aMin = 5; aMax = 80; da = 1;
SetDelayed[ ... ]; aC = Table[ ... ]; aN = Table[ ... ]; r = Table[ ... ];
TableView[ { ... } ]
```



Selbst ohne das die Mortalität potenzierende Problem der Überlastung des Gesundheitssystems stellt das SARS - CoV - 2 Virus stellt somit eine relevante Bedrohung für die Bevölkerungsgruppe der über 50 Jährigen dar.

Die leichte Übertragbarkeit und der große Anteil an stillen Übertragungen macht es zudem erforderlich, die zweite Kategorie von Einflussfaktoren heranzuziehen.

6.3.2 Populationseigenschaften und Verhalten

Ist die Reproduktionszahl eines Erregers größer als eins, so stellte sich die Frage, welche (aus Sicht der Medizin) externen Faktoren den Verlauf einer Epidemie beeinflussen können. Wie wir in Abschnitt 2 gesehen haben, wird der Verlauf einer Epidemie ($R(t)$) von der Frequenz infektiöser Kontakte bestimmt, wobei es über die Identifikation der Erkrankten möglich ist, diese zu verringern. Alle der folgenden Faktoren überlagern sich und ergeben das für den Verlauf relevante $R(t)$. Für die Geschwindigkeit des Verlaufes ist besonders die anfängliche Reproduktionszahl R_0 bedeutsam.

- Jede Population hat eine kulturell, demografisch und jahreszeitlich begründete Kontaktfrequenz, die R_0 prägt.
- Gesellschaften können sich auf Epidemien vorbereiten und für eine Infrastruktur sorgen, die eine Epidemie schnell eindämmen kann, z.B. durch Tests und Quarantänemaßnahmen. Den Parameter q_c , der die Wahrscheinlichkeit angibt, mit der eine Neuinfektion erkannt und die entsprechende Person an der Weiterverbreitung der Erreger gehindert wird ist nichts anderes als eine Zahl zwischen 0 (100% effektive Maßnahmen) und 1 (keine Maßnahmen) mit der dann R zu multiplizieren ist.

- Die aufwendigste Maßnahme ist der Shutdown, d.h. eine maximale Beeinflussung der Epidemie durch Verhinderung von infektiösen Kontakten.

Eine Pandemie wird auch durch die Interaktion von Populationen bestimmt (Reisebeschränkungen). Die einzelnen $R(t)$ Verläufe werden wiederum durch die (oft erratisch erscheinenden) Maßnahmen innerhalb der Populationen bestimmt. Somit ist es selbst im Nachhinein sehr schwierig die Effizienz von Maßnahmen zu quantifizieren. Gleichzeitig zeigen bereits einfache Modelle, wie das Systemverhalten optimal beeinflusst werden kann. Das fortgesetzte Emittieren von CO_2 wurde von Elon Musk als das blödeste Experiment aller Zeiten bezeichnet. Pandemien ermöglichen ähnlich sinnlose Experimente, die jedoch einfacher erfahrbar sind.

6.3.3 Wann soll eine Beeinflussung beginnen?

Auf diese Frage könnte es eine einfache Antwort geben, die jedoch nur schwer umsetzbar ist: Sobald die Freisetzung eines hinreichend gefährlichen Erregers bemerkt wird, sollte sofort mit allen wirksamen Maßnahmen (die zu $R < 1$ führen) reagiert werden. Kompromisse verschlechtern das Interventionsergebnis und dann ist ein Ende mit Schrecken ist besser als ein Schrecken ohne Ende!?

6.3.4 Wann darf eine Beeinflussung enden?

Sie können es nun selbst nachrechnen: Erst wenn die Infektion nicht mehr weitergegeben werden kann, ist die Epidemie zu Ende. Werden die wirksamen Eindämmungsmaßnahmen früher beendet, so kommt (mit einer gewissen Latenzzeit) die nächste Welle.

6.3.5 Gibt es eine optimale Vorgehensweise?

Wenn ein möglichst frühzeitiges und bis zum letzten Infizierten durchhaltendes Eingreifen die optimale Vorgehensweise ist, so gibt es eigentlich nicht mehr viel zu optimieren. Die verschiedenen Interventionen bedeuten jedoch sicherlich einen mit sinkendem **RI** stark steigenden Aufwand. Nur wenn sich die Intervention bis zum letzten Infizierten durchhalten lässt, scheint dieser Aufwand gerechtfertigt zu sein.

Wenn wir jedoch annehmen, das bei allen Interventionen immer Infizierte (genauer Infektiöse) übrig bleiben, so können wir versuchen, ein optimales **RI** zu berechnen. Dazu kann das Werkzeug **MinimizeEpidemicOutcome[]** dienen:

In[]:= CompoundExpression[...]

Für

RI	0.1
R	4.0
n0	10 000
xdImin	0.00002
xdImax	0.01
fLearn	1.0
nIO	10
tMax	50.0
fRMaxOutcome	1000.0
fIMaxOutcome	2.0
ftMaxOutcome	10.0
fRIOOutcome	10.0

beträgt das optimale RI 0.195571

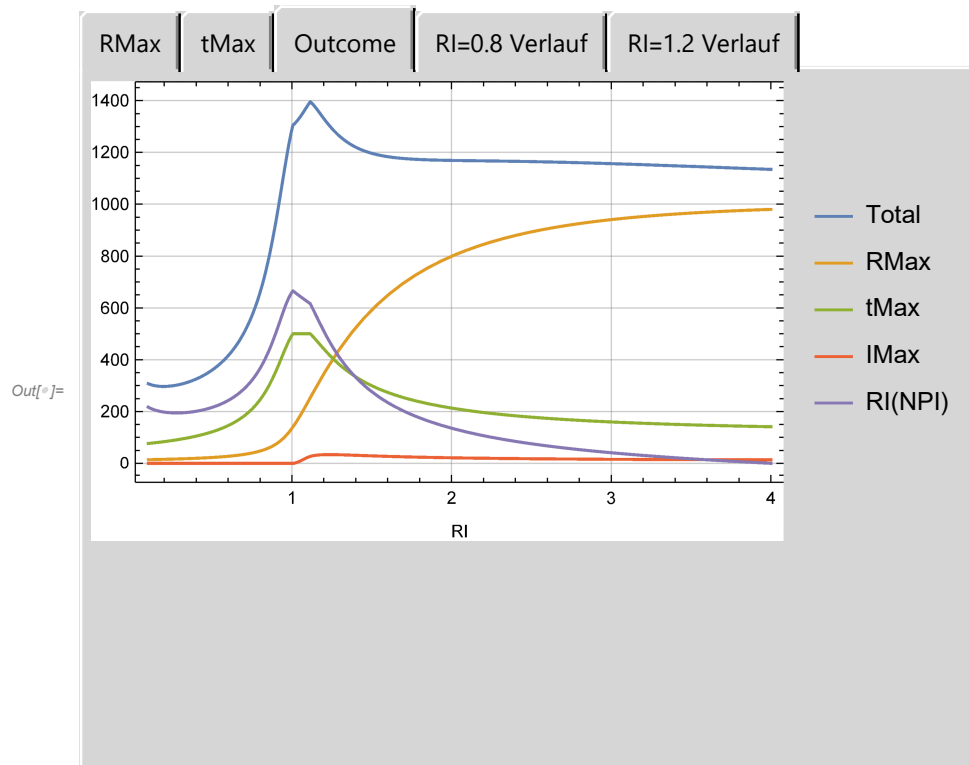
Da es jedoch hier mehr als ein Ziel zu erreichen gilt hängt das “optimale” Ergebnis immer von den Gewichtungen der Einzelziele und natürlich von dem Parameter **xdImin** ab, der entscheidet, ob es weitere Wellen gibt. Hier hilft eine Betrachtung der Abhängigkeit der Ergebnisse von der Interventionsintensität **RI**:

In[]:= CompoundExpression[...]

ClearAll[f];
f[ri_?NumericQ] := f[ri] = EpidemicOutcome[Join[{"RI" → ri}, inp]];

SetOptions[...];

TabView[...];



Da der Anteil der letztendlich von der Epidemie Betroffenen **RMax** für **RI**>1 schnell ansteigt, macht die ganze Optimierung nur Sinn, wenn diese Größe kein hohes Gewicht hat oder ein **RI** kleiner als 1 nicht erreichbar ist. Ansonsten ist eine maximale Intervention (**RI** deutlich unter 1) bis zum letzten Infizierten durchzuhalten. Da dies mit zunehmender Populationsgröße immer schwieriger wird ist eine Aufteilung in Teilpopulation, die sich untereinander nicht anstecken können, wichtig.

7. Zusammenfassung

Natürlich enttäuscht der Befund, dass der genaue (quantitative) Verlauf einer Epidemie nicht wirklich vorhersagbar ist [7]. Es lässt sich jedoch ein viel besseres Ergebnis anpeilen: Wenn spürbare Schutzmaßnahmen von allen akzeptiert werden, bevor die Epidemie selbst für jeden erfahrbar ist, dann können Früherkennungsmethoden Epidemien auch wirksam verhindern. Auch für diese Frühwarnsysteme ist eine Entwicklung hin zu immer genauer werdenden Modellen (wie bei der Wettervorhersage) erforderlich.

Durchaus enttäuschend ist die Menge an Details die schon bei einem so einfach strukturiertem Problem zur Erfassung der wesentlichen Aspekte notwendig sind. *There is no free lunch.*

Überraschend ist die Qualität und Menge an Schlussfolgerungen die sich aus einem Modell ableiten lassen, das gar keine konkrete Vorhersage von kommenden Ereignissen bezweckte und daher eher eine Parabel wie Daisyworld [5] darstellt.

Das Erlernen der Fertigkeit der Prozessmodellierung erfordert einen Erwerb von Fähigkeiten im Bereich des wissenschaftlichen Arbeitens und des *computational thinking*. Mit einem entsprechenden Werkzeugkasten und eigenen Computerexperimenten lassen sich nicht triviale Prozesse besser verstehen, nicht selten sogar besser als auf der Basis eigener praktischer “Erfahrungen”, die nicht selten eher zu emotional bedingten als zu Evidenz basierten Schlussfolgerungen führen.

Ganz wesentlich ist der neue qualitative Aspekt von rechnenden Dokumenten wie diesem: Hier können sie als Leser nicht nur den Standpunkt der Autoren einfacher nachprüfen sondern sich auch selbst eigene Ansichten des Prozesses erarbeiten.

Schlussbemerkung

In eine wissenschaftliche Veröffentlichung gehören sicher weitere Hinweise auf den Stand der Forschung und die wissenschaftlichen Prioritäten (wer hat was zuerst publiziert). Zudem sollten auch theoretische Arbeiten einen Bezug zu realen Beobachtungen und Messungen herstellen. Hier lag der Fokus auf einem für Mensch und Computer verständlichen Dokument, welches den Nutzen einer computational language [8] als Sprache der Wissenschaft demonstriert: Sie als Leser können die Rechnungen nicht nur nachvollziehen, sondern für alle Modelle selbst das Universum der Möglichkeiten erkunden und eigene “Entdeckungen” machen.

Literatur

1. F Brauer et.al. (2008):
Mathematical Epidemiology.
Springer, Berlin
DOI:10.1007/978-3-540-78911-6
2. G S Callendar (1938):
The artificial production of carbon dioxide and its influence on temperature.
Quarterly Journal of the Royal Meteorological Society **64** (1938) 223,
DOI:10.1002/qj.49706427503
3. S H Schneider, R Londer (1984):
The coevolution of climate and life.
Sierra Club Books, San Francisco
4. P Bauer, A Thorpe, G Brunet (2015):
The quiet revolution of numerical weather prediction
Nature **525** (2015) 47,
DOI:10.1038/nature14956
5. A J Watson, J E Lovelock (1983):
Biological homeostasis of the global environment: the parable of Daisyworld.
Tellus **35 B** (1983) 284,
DOI:10.3402/tellusb.v35i4.14616
6. W O Kermack, A McKendrick (1927):
A contribution to the Mathematical Theory of Epidemics.
Proc. Roy. Soc. Lond. **A115** (1927) 700,
DOI:10.1098/rspa.1927.0118
7. J P Ioannidis, S Cripps, M A Tanner (2020):
Forecasting for COVID-19 has failed.
International Journal of Forecasting (2020) in press,
DOI:10.1016/j.ijforecast.2020.08.004
8. S Wolfram (2019):
What we've built is a computational language and that's very important.
<https://writings.stephenwolfram.com/2019/05/what-weve-built-is-a-computational-language-and-thats-very-important/>
9. J Wendelstorf (2019):
The accuracy of indirect measurements and model calibration by falsification.
DOI: 10.21268/20190515-4
10. G Calvet et.al. (2020):
Estimates of the severity of coronavirus disease 2019: a model-based analysis.
The Lancet Infectious Diseases **20** (2020) 669,
DOI: 10.1016/S1473-3099(20)30243-7

Revisionsgeschichte

01.10.2020 [J Wendelstorf] Die erste publizierte Version.

Schlussbemerkung

Der einflussreiche Ökonom John Maynard Keynes sagte *When I'm wrong, I change my mind. What do you do?*

Derartige Dokumente sollten viele Revisionen durchlaufen. Die Verbesserungen bringen dann weitere Autoren in die Liste.

Wenn Sie dieses Dokument studiert haben, dann möchten Sie demnächst vielleicht selbst ein *computational document* verfassen und so den transparenten Fortschritt Ihrer wissenschaftlichen Disziplin unterstützen.

Hoffentlich haben Sie etwas über Modelle gelernt und erkennen den Nutzen von *computational documents*.